aws

Guía para desarrolladores

Amazon Simple Queue Service

Amazon Simple Queue Service: Guía para desarrolladores

Table of Contents

¿Qué es Amazon Simple Queue Service?

Con Amazon Simple Queue Service (Amazon SQS), se ofrece una cola alojada segura, duradera y disponible que le permite integrar y desacoplar sistemas y componentes de software distribuidos. Amazon SQS ofrece constructos comunes, como colas de mensajes fallidos y etiquetas de asignación de costos. Proporciona una API de servicios web genérica a la que puede acceder mediante cualquier lenguaje de programación compatible con el AWS SDK.

Temas

Beneficios de utilizar Amazon SQS

- Seguridad: usted controla quién puede enviar mensajes a la cola de Amazon SQS o recibirlos de ella. Puede elegir entre transmitir datos confidenciales mediante la protección del contenido de los mensajes en las colas por medio del cifrado del servidor (SSE) administrado por Amazon SQS de forma predeterminada o mediante el uso de claves SSE personalizadas administradas en AWS Key Management Service (AWS KMS).
- Durabilidad: para mantener seguros sus mensajes, Amazon SQS los almacena en varios servidores. Las colas estándar admiten la entrega de at-least-once mensajes y las colas FIFO admiten el procesamiento de mensajes exactamente una vez y el modo de alto rendimiento.
- Disponibilidad: Amazon SQS utiliza una infraestructura redundante para proporcionar acceso con un alto grado de simultaneidad a los mensajes y alta disponibilidad para producir y consumir mensajes.
- Escalabilidad: Amazon SQS puede procesar cada solicitud en búfer independientemente, con un escalado transparente para controlar cualquier aumento o pico de carga sin instrucciones de aprovisionamiento.
- Fiabilidad: Amazon SQS bloquea sus mensajes durante el procesamiento, de forma que varios productores puedan enviar mensajes y varios consumidores puedan recibirlos al mismo tiempo.
- Personalización: sus colas no tienen por qué ser exactamente iguales; por ejemplo, puede establecer un retraso predeterminado en una cola. Puede almacenar el contenido de los mensajes con un tamaño superior a 256 KB mediante Amazon Simple Storage Service (Amazon S3) o

Amazon DynamoDB (donde Amazon SQS mantiene un puntero que señala al objeto de Amazon S3) o bien puede dividir los mensajes grandes en mensajes más pequeños.

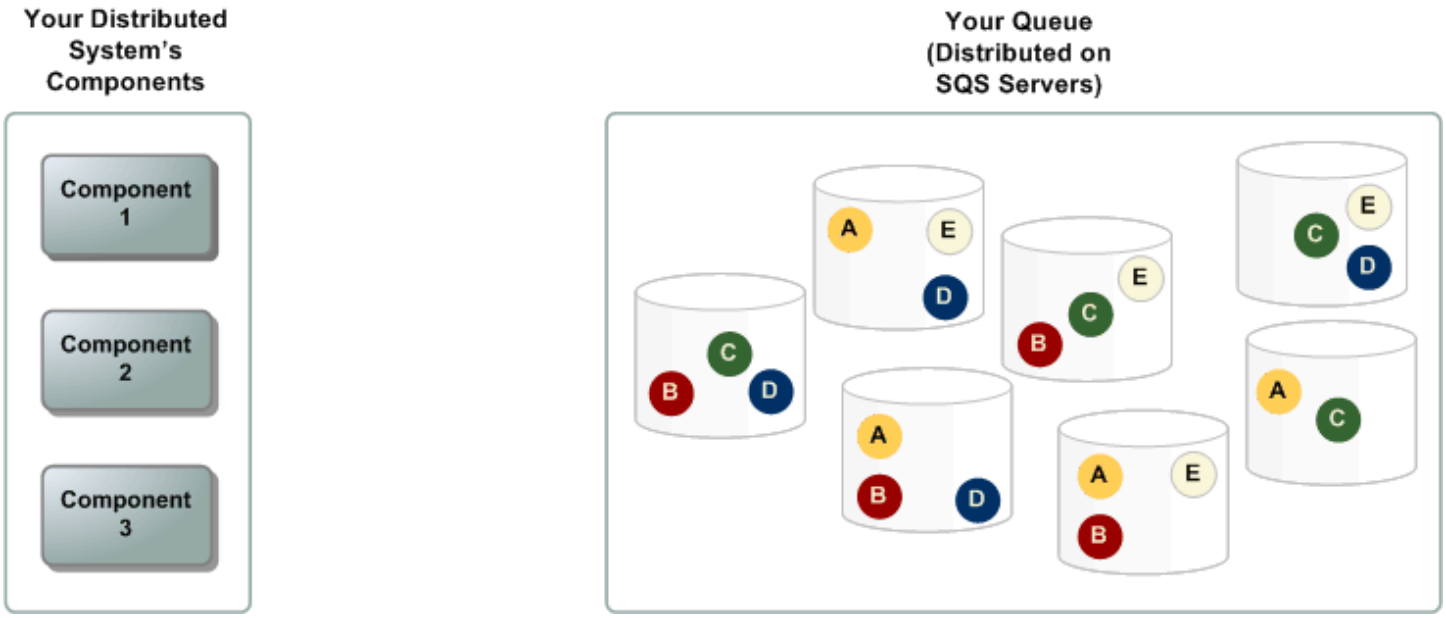
Arquitectura de Amazon SQS básica

En esta sección se describen las partes de un sistema de mensajería distribuido y se explica el ciclo de vida de un mensaje de Amazon SQS.

Colas distribuidas

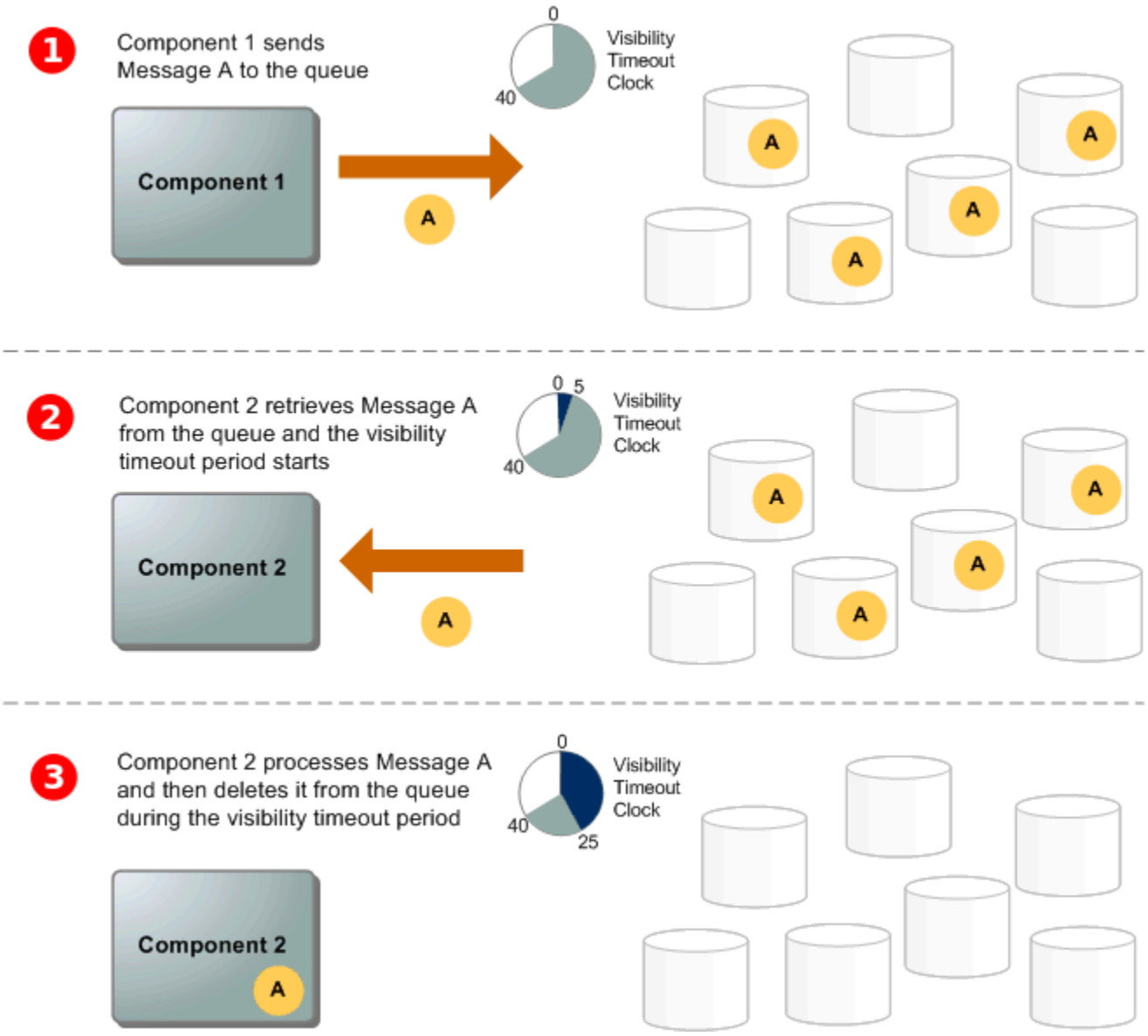
Un sistema de mensajería distribuido consta de tres partes principales: los componentes del sistema distribuido, la cola (distribuida en los servidores de Amazon SQS) y los mensajes de la cola.

En el siguiente escenario, el sistema tiene varios productores (componentes que envían mensajes a la cola) y consumidores (componentes que reciben mensajes de la cola). La cola (que contiene los mensajes A a E) almacena de forma redundante los mensajes en varios servidores de Amazon SQS.



Ciclo de vida del mensaje

En el escenario siguiente se describe el ciclo de vida de un mensaje de Amazon SQS en una cola, desde la creación hasta la eliminación.



1

Un productor (componente 1) envía el mensaje A a la cola y el mensaje se distribuye entre los servidores de Amazon SQS de forma redundante.

2

Cuando un consumidor (componente 2) está preparado para procesar mensajes, consume mensajes de la cola y se devuelve el mensaje A. Mientras se procesa, el mensaje A permanece en la cola y no se devuelve a las solicitudes de recepción posteriores durante el tiempo de espera de visibilidad.

3

El consumidor (componente 2) elimina el mensaje A de la cola para evitar que se reciba y procese de nuevo cuando termina el tiempo de espera de visibilidad.

> **Note**
>
> Amazon SQS elimina automáticamente los mensajes que han estado en una cola durante más tiempo que el periodo máximo de retención de mensajes. El periodo de retención de mensajes predeterminado es de 4 días. Sin embargo, puede establecer el periodo de retención de un mensaje en un valor comprendido entre 60 y 1.209.600 segundos (14 días) mediante la acción `SetQueueAttributes` .

Diferencias entre Amazon SQS, Amazon MQ y Amazon SNS

Amazon SQS, Amazon SNS y Amazon MQ ofrecen servicios de mensajería gestionados easy-to-use y altamente escalables, cada uno diseñado para funciones específicas dentro de los sistemas distribuidos. A continuación, se ofrece una descripción general mejorada de las diferencias entre estos servicios:

Amazon SQS desacopla y escala los sistemas y componentes de software distribuidos como un servicio de colas. Por lo general, procesa los mensajes a través de un único suscriptor, lo que resulta ideal para flujos de trabajo en los que la prevención de pedidos y pérdidas es fundamental. Para una distribución más amplia, la integración de Amazon SQS con Amazon SNS permite un patrón de mensajería distribuido, que envía mensajes de manera efectiva a varios suscriptores a la vez.

Amazon SNS permite a los editores enviar mensajes a varios suscriptores a través de temas, que sirven como canales de comunicación. Los suscriptores reciben los mensajes publicados mediante un tipo de punto final compatible Amazon Data Firehose, como Amazon SQS, Lambda, HTTP, correo electrónico, notificaciones push móviles y mensajes de texto (SMS) móviles. Este servicio es ideal para situaciones que requieren notificaciones inmediatas, como la participación de los usuarios en tiempo real o los sistemas de alarma. Para evitar la pérdida de mensajes cuando los suscriptores están desconectados, la integración de Amazon SNS con los mensajes en cola de Amazon SQS garantiza una entrega uniforme.

Amazon MQ se adapta mejor a las empresas que desean migrar desde los intermediarios de mensajes tradicionales, ya que admite protocolos de mensajería estándar como AMQP y MQTT,

junto con Apache ActiveMQ y RabbitMQ. Ofrece compatibilidad con los sistemas heredados que necesitan una mensajería estable y fiable sin necesidad de una reconfiguración significativa.

El siguiente cuadro proporciona una descripción general del tipo de recurso de cada servicio:

Tipo de recurso	Amazon SNS	Amazon SQS	Amazon MQ
Síncrono	No	No	Sí
Asíncrono	Sí	Sí	Sí
Queues	No	Sí	Sí
Mensajería de publicador y suscriptor	Sí	No	Sí
Agentes de mensajes	No	No	Sí

Se recomiendan tanto Amazon SQS como Amazon SNS para nuevas aplicaciones que puedan beneficiarse de una escalabilidad casi ilimitada y de API sencillas. Por lo general, con sus pay-as-you-go precios, ofrecen soluciones más rentables para aplicaciones de gran volumen. Recomendamos Amazon MQ para migrar aplicaciones de los intermediarios de mensajes existentes que dependen de la compatibilidad con API como JMS o protocolos como el Protocolo avanzado de colas de mensajes (AMQP), MQTT y el Protocolo simple de mensajes orientados a texto (OpenWireSTOMP).

Configuración de Amazon SQS

Para poder usar Amazon SQS por primera vez, debe completar los pasos siguientes.

Temas

Paso 1: Crear un usuario Cuenta de AWS de IAM

Para acceder a cualquier AWS servicio, primero debes crear una Cuenta de AWScuenta de Amazon.com que pueda usar AWS productos. Puedes utilizarla Cuenta de AWS para ver tus informes de actividad y uso y para gestionar la autenticación y el acceso.

Para evitar utilizar el usuario Cuenta de AWS raíz para las acciones de Amazon SQS, se recomienda crear un usuario de IAM para cada persona que necesite acceso administrativo a Amazon SQS.

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .
2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Paso 2: Conceder acceso mediante programación

Para utilizar las acciones de Amazon SQS (por ejemplo, mediante Java o a través de AWS Command Line Interface), necesita un identificador de clave de acceso y una clave de acceso secreta.

> **Note**
>
> El identificador de la clave de acceso y la clave de acceso secreta son específicos de AWS Identity and Access Management. No las confunda con las credenciales de otros AWS servicios, como los pares de claves de Amazon EC2.

Los usuarios necesitan acceso programático si quieren interactuar con personas AWS ajenas a. AWS Management Console La forma de conceder el acceso programático depende del tipo de usuario que acceda. AWS

Para conceder acceso programático a los usuarios, elija una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal (Usuarios administrados en el IAM Identity Center)	Usa credenciales temporales para firmar las solicitudes programáticas a los AWS CLI AWS SDK o las API. AWS	Siga las instrucciones de la interfaz que desea utilizar: • Para ello AWS CLI, consulte Configuración del uso AWS IAM Identity Center en AWS CLI la Guía del AWS Command Line Interface usuario. • Para ver AWS los SDK, las herramientas y las AWS API, consulte la autenticación del IAM Identity Center en la Guía de referencia de AWS los SDK y las herramientas.
IAM	Utilice credenciales temporales para firmar las solicitudes programáticas a los AWS SDK o las AWS CLI API. AWS	Siga las instrucciones de Uso de credenciales temporales con AWS recursos de la Guía del usuario de IAM.
IAM	(No recomendado) Utilice credenciales de larga duración para firmar las solicitudes programáticas a los AWS CLI AWS SDK o las API. AWS	Siga las instrucciones de la interfaz que desea utilizar: • Para ello AWS CLI, consulte Autenticación con credenciales de usuario de IAM en la Guía del usuario.AWS Command Line Interface • Para obtener información AWS sobre los SDK y las herramientas, consulte Autenticarse con credencia

¿Qué usuario necesita acceso programático?	Para	Mediante
		les de larga duración en la Guía de referencia de los AWS SDK y las herramien tas. • Para obtener informaci ón AWS sobre las API, consulte Administrar las claves de acceso para los usuarios de IAM en la Guía del usuario de IAM.

Paso 3: prepararse para usar el código de ejemplo

Esta guía incluye ejemplos que utilizan el AWS SDK for Java. Para ejecutar el código de ejemplo, siga las instrucciones de configuración de Introducción a AWS SDK para Java 2.0.

Puede desarrollar AWS aplicaciones en otros lenguajes de programación, como GoJavaScript, Python y Ruby. Para obtener más información, consulte Herramientas sobre las que construir AWS.

> **Note**
>
> Puede explorar Amazon SQS sin necesidad de escribir código con herramientas como AWS Command Line Interface (AWS CLI) o Windows. PowerShell Puede encontrar AWS CLI ejemplos en la sección Amazon SQS de la Referencia de AWS CLI comandos. Puede encontrar PowerShell ejemplos de Windows en la sección Amazon Simple Queue Service de la referencia de AWS Tools for PowerShell cmdlets.

Siguientes pasos

Ya está preparado para empezar a administrar las colas y los mensajes de Amazon SQS mediante la AWS Management Console.

Introducción a Amazon SQS

En esta sección, aprenderá a crear colas estándar o FIFO con la consola Amazon SQS.

Temas

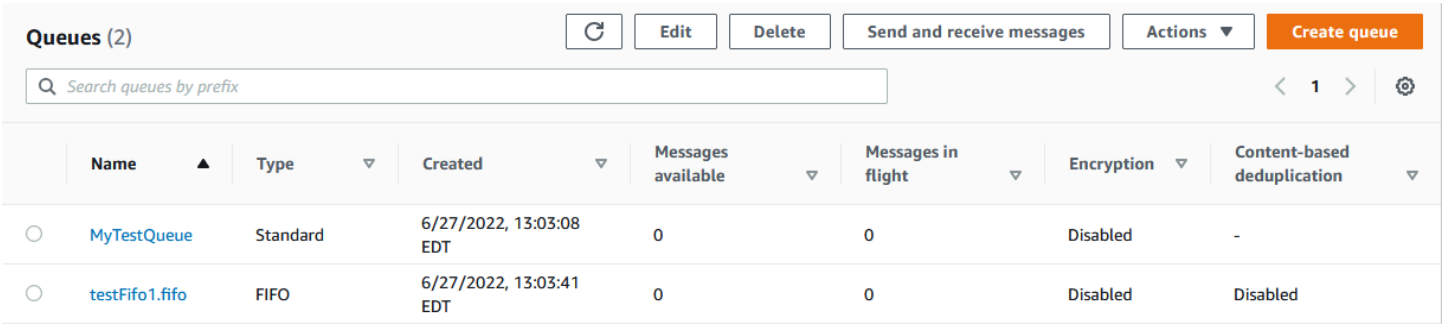
Requisitos previos

Antes de comenzar, complete los pasos de Configuración de Amazon SQS.

Descripción de la consola de Amazon SQS

Cuando abra la consola Amazon SQS, seleccione Coleas en el panel de navegación. La página Colas proporciona información sobre todas las colas de la región activa.

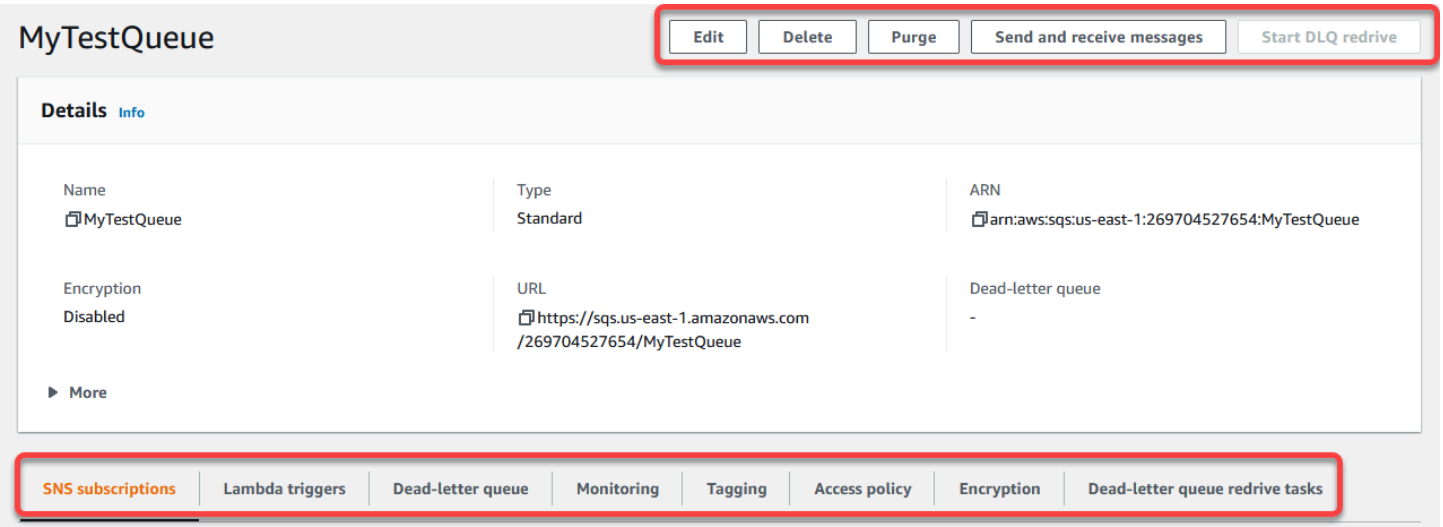
Cada entrada de cola proporciona información esencial sobre la cola, incluidos su tipo y sus atributos clave. Las colas estándar, optimizadas para obtener el máximo rendimiento y ordenar los mensajes de la mejor manera, se diferencian de las colas FIFO (primero en entrar, primero en salir), que priorizan el orden y la exclusividad de los mensajes para las aplicaciones que requieren una secuenciación estricta de los mensajes.



Elementos y acciones interactivos

En la página Colas, tienes varias opciones para gestionar tus colas:

1. Acciones rápidas: junto al nombre de cada cola, hay un menú desplegable que ofrece un acceso rápido a las acciones más habituales, como enviar mensajes, ver o eliminar mensajes, configurar los activadores y eliminar la propia cola.
2. Vista detallada y configuración: al hacer clic en el nombre de una cola, se abre su página de detalles, donde puede profundizar en los ajustes y configuraciones de la cola. Aquí puede ajustar parámetros como el período de retención de los mensajes, el tiempo de espera de visibilidad y el tamaño máximo de los mensajes para adaptar la cola a los requisitos de su aplicación.



Selección de regiones y etiquetas de recursos

Asegúrese de estar en lo correcto Región de AWS para acceder a sus colas y gestionarlas de forma eficaz. Además, considere la posibilidad de utilizar etiquetas de recursos para organizar y categorizar las colas, lo que permitirá una mejor administración de los recursos, la asignación de costos y el control del acceso en su AWS entorno compartido.

Al aprovechar las características y funcionalidades que ofrece la consola Amazon SQS, puede gestionar de forma eficiente su infraestructura de mensajería, optimizar el rendimiento de las colas y garantizar una entrega fiable de los mensajes para sus aplicaciones.

Tipos de colas de Amazon SQS

Amazon SQS admite dos tipos de colas: estándar y FIFO. Utilice la información de la tabla siguiente para elegir la cola adecuada a su situación. Para obtener más información sobre las colas de Amazon SQS, consulte Introducción a las colas estándar de Amazon SQS y Introducción a las colas FIFO en Amazon SQS.

Colas estándar	Colas FIFO
Rendimiento ilimitado: las colas estándar admiten un número casi ilimitado de llamadas a la API por segundo, por acción de API (`SendMessage` , `ReceiveMessage` o `DeleteMessage`). Al menos una entrega: se envía un mensaje al menos una vez, pero ocasionalmente se entrega más de una copia del mensaje. Orden de mejor esfuerzo: ocasionalmente, los mensajes pueden entregarse en un orden distinto al que se enviaron.	Alto rendimiento: si utiliza el procesamiento por lotes, las colas FIFO admiten hasta 3000 mensajes por segundo y por método de API (`SendMessageBatch` , `ReceiveMessage` o `DeleteMessageBatch`). Los 3000 mensajes por segundo representan 300 llamadas a la API, cada una con un lote de 10 mensajes. Para solicitar un aumento de la cuota, envíe una solicitud de soporte técnico. Sin lotes, las colas FIFO admiten hasta 300 llamadas a la API por segundo, por método API (`SendMessage` , `ReceiveMessage` o `DeleteMessage`). Procesamiento único: un mensaje se entrega una vez y permanece disponible hasta que el cliente lo procesa y elimina. No se introducen duplicados en la cola. Entrega primero en entrar, primero en salir: el orden en que los mensajes se envían y se reciben se conserva estrictamente.
	
Envíe datos entre aplicaciones cuando el desempeño sea importante, por ejemplo: • Desacoplar solicitudes de usuario en tiempo real de un trabajo en segundo plano intensivo: permitir a los usuarios cargar archivos multimedia mientras cambia el tamaño o se codifican.	Envíe datos entre aplicaciones cuando el orden de los eventos sea importante, por ejemplo: • Asegúrese de que los comandos introducidos por el usuario se ejecutan en el orden correcto.

Colas estándar	Colas FIFO
• Asignar tareas para múltiples nodos de trabajo: procesar un elevado número de solicitudes de validación de tarjetas de crédito. • Agrupar mensajes para procesarlos más adelante: programar varias entradas para añadirlas a una base de datos.	• Para mostrar el precio correcto del producto enviando las modificaciones de precios en el orden adecuado. • Para evitar que un estudiante se matricule en un curso antes de registrarse para obtener una cuenta.

Creación de una cola estándar de Amazon SQS y envío de un mensaje

El modo de crear una cola estándar para Amazon SQS es el siguiente.

Cree una cola con la consola Amazon SQS

Puede usar la consola de Amazon SQS para crear colas estándar. La consola proporciona valores predeterminados para todas las configuraciones excepto para el nombre de la cola.

> **Important**
>
> El 17 de agosto de 2022, se aplicó de forma predeterminada el cifrado del servidor (SSE) a todas las colas de Amazon SQS.
> No agregue información de identificación personal (PII) ni ninguna otra información confidencial o sensible en los nombres de las colas. Muchos Amazon Web Services pueden acceder a los nombres de las colas, incluidos los CloudWatch registros y la facturación. Los nombres de las colas no están diseñados para contener información privada o confidencial.

Creación de una cola estándar de Amazon SQS

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. Elija Crear cola.
3. En Tipo, el tipo de cola estándar se establece de forma predeterminada.

> **Note**
>
> No se puede cambiar el tipo de cola después de crearla.

4. Escriba un Nombre para la cola.
5. (Opcional) La consola establece valores predeterminados para los parámetros de configuración de la cola. En Configuración, puede establecer nuevos valores para los siguientes parámetros:

   a. En Tiempo de espera de visibilidad, introduzca la duración y las unidades. El intervalo es de 0 segundos a 12 horas. El valor de predeterminado es de 30 segundos.

   b. En Periodo de retención del mensaje, introduzca la duración y las unidades. El intervalo es de 1 minuto a 14 días. El valor predeterminado es 4 días.

   c. En Retraso de entrega, introduzca la duración y las unidades. El intervalo es de 0 segundos a 15 minutos. El valor predeterminado es 0 segundos.

   d. En Tamaño máximo del mensaje, introduzca un valor. El intervalo es de 1 KB a 256 KB. El valor predeterminado es 256 KB.

   e. En Tiempo de espera de recepción del mensaje, introduzca un valor. El intervalo es de 0 a 20 segundos. El valor predeterminado es 0 segundos, lo que establece un sondeo corto. Cualquier valor distinto de cero establece un sondeo largo.

6. (Opcional) Defina una política de acceso. La política de acceso define las cuentas, los usuarios y los roles que pueden acceder a la cola. La política de acceso también define las acciones (como `SendMessage`, `ReceiveMessage` o`DeleteMessage`) a las que pueden acceder los usuarios. La política predeterminada permite que solo el propietario de la cola envíe y reciba mensajes.

   Para definir la política de acceso, realice una de las siguientes acciones:

   - Elija Básico para configurar quién puede enviar mensajes a la cola y quién puede recibir mensajes de la cola. La consola crea la política basándose en sus elecciones y muestra la política de acceso resultante en el panel JSON de solo lectura.
   - Elija Avanzado para modificar directamente la política de acceso de JSON. De este modo, puede especificar un conjunto personalizado de acciones que puede realizar cada entidad principal (cuenta, usuario o rol).

7. En Política de permiso de redireccionamiento, elija Habilitado. Seleccione una de las siguientes opciones: Permitir todo, Por cola o Denegar todo. Al elegir Por cola, especifique una lista de hasta diez colas de origen por nombre de recurso de Amazon (ARN).

8. De forma predeterminada, Amazon SQS proporciona un cifrado del servidor administrado. Para elegir un tipo de clave de cifrado o para desactivar el cifrado del servidor administrado por Amazon SQS, expanda Cifrado. Para obtener más información sobre los tipos de claves de cifrado, consulte Configuración del cifrado del lado del servidor para una cola mediante claves de cifrado administradas por SQL y Configuración del cifrado del lado del servidor para una cola mediante la consola Amazon SQS.

   > **Note**
   >
   > Con SSE activado, se rechazarán las solicitudes `SendMessage` y `ReceiveMessage` anónimas a la cola cifrada. Las prácticas recomendadas de seguridad de Amazon SQS desaconsejan utilizar solicitudes anónimas. Si desea enviar solicitudes anónimas a una cola de Amazon SQS, asegúrese de desactivar SSE.

9. (Opcional) Para configurar una cola de mensajes fallidos a fin de recibir mensajes que no se pueden entregar, expanda Cola de mensajes fallidos.
10. (Opcional) Para agregar etiquetas a la cola, expanda Etiquetas.
11. Elija Crear cola. Amazon SQS crea la cola y muestra la página Detalles de la cola.

Amazon SQS propaga la información sobre la nueva cola por todo el sistema. Dado que Amazon SQS es un sistema distribuido, es posible que experimente un ligero retraso antes de que la consola muestre la cola en la página Colas.

Enviar un mensaje

Después de crear la cola, puede enviarle un mensaje.

1. En el panel de navegación izquierdo, elija Colas. En la lista de colas, seleccione la cola que ha creado.
2. En Acciones, elija Enviar y recibir mensajes.

   La consola muestra la página Enviar y recibir mensajes.
3. En Cuerpo del mensaje, especifique el texto del mensaje.
4. Para una cola estándar, puede introducir un valor para Retraso de entrega y elegir las unidades. Por ejemplo, introduzca `60` y elija segundos. Para obtener más información, consulte Temporizadores de mensajes de Amazon SQS.
5. Elija Enviar mensaje.

Cuando se envía el mensaje, la consola muestra un mensaje de confirmación. Elija Ver detalles para mostrar información sobre el mensaje enviado.

Creación de una cola FIFO de Amazon SQS y envío de un mensaje

El modo de crear una cola FIFO para Amazon SQS es el siguiente.

Creación de una cola

Puede usar la consola de Amazon SQS para crear colas FIFO. La consola proporciona valores predeterminados para todas las configuraciones excepto para el nombre de la cola.

> **Important**
>
> El 17 de agosto de 2022, se aplicó de forma predeterminada el cifrado del servidor (SSE) a todas las colas de Amazon SQS.
> No agregue información de identificación personal (PII) ni ninguna otra información confidencial o sensible en los nombres de las colas. Muchos Amazon Web Services pueden acceder a los nombres de las colas, incluidos los CloudWatch registros y la facturación. Los nombres de las colas no están diseñados para contener información privada o confidencial.

Creación de una cola FIFO de Amazon SQS

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. Elija Crear cola.
3. En Tipo, el tipo de cola estándar se establece de forma predeterminada. Para crear una cola FIFO, elija FIFO.

   > **Note**
   >
   > No se puede cambiar el tipo de cola después de crearla.

4. Escriba un Nombre para la cola.

La cola FIFO debe finalizar con el sufijo `.fifo`. El sufijo cuenta para la cuota de nombre de cola de 80 caracteres. Para determinar si una cola es FIFO, puede comprobar si el nombre de la cola termina con el sufijo.

5. (Opcional) La consola establece valores predeterminados para los parámetros de configuración de la cola. En Configuración, puede establecer nuevos valores para los siguientes parámetros:

   a. En Tiempo de espera de visibilidad, introduzca la duración y las unidades. El intervalo es de 0 segundos a 12 horas. El valor de predeterminado es de 30 segundos.

   b. En Periodo de retención del mensaje, introduzca la duración y las unidades. El intervalo es de 1 minuto a 14 días. El valor predeterminado es 4 días.

   c. En Retraso de entrega, introduzca la duración y las unidades. El intervalo es de 0 segundos a 15 minutos. El valor predeterminado es 0 segundos.

   d. En Tamaño máximo del mensaje, introduzca un valor. El intervalo es de 1 KB a 256 KB. El valor predeterminado es 256 KB.

   e. En Tiempo de espera de recepción del mensaje, introduzca un valor. El intervalo es de 0 a 20 segundos. El valor predeterminado es 0 segundos, lo que establece un sondeo corto. Cualquier valor distinto de cero establece un sondeo largo.

   f. Para una cola FIFO, elija Desduplicación basada en el contenido para activar la desduplicación basada en el contenido. La configuración predeterminada está desactivada.

   g. (Opcional) Para que una cola FIFO permita un mayor rendimiento en el envío y recepción de mensajes en la cola, elija Habilitar FIFO de alto rendimiento.

      Al elegir esta opción, las opciones relacionadas (Ámbito de desduplicación y Límite de rendimiento FIFO) cambian a la configuración necesaria para habilitar un alto rendimiento para las colas FIFO. Si cambia alguna de las configuraciones necesarias para utilizar FIFO de alto rendimiento, el rendimiento normal estará en vigor para la cola y la desduplicación se producirá según lo especificado. Para obtener más información, consulte Alto rendimiento de las colas FIFO en Amazon SQS y Cuotas de mensajes de Amazon SQS.

6. (Opcional) Defina una política de acceso. La política de acceso define las cuentas, los usuarios y los roles que pueden acceder a la cola. La política de acceso también define las acciones (como `SendMessage`, `ReceiveMessage` o`DeleteMessage`) a las que pueden acceder los usuarios. La política predeterminada permite que solo el propietario de la cola envíe y reciba mensajes.

   Para definir la política de acceso, realice una de las siguientes acciones:

- Elija Básico para configurar quién puede enviar mensajes a la cola y quién puede recibir mensajes de la cola. La consola crea la política basándose en sus elecciones y muestra la política de acceso resultante en el panel JSON de solo lectura.
- Elija Avanzado para modificar directamente la política de acceso de JSON. De este modo, puede especificar un conjunto personalizado de acciones que puede realizar cada entidad principal (cuenta, usuario o rol).

7. En Política de permiso de redireccionamiento, elija Habilitado. Seleccione una de las siguientes opciones: Permitir todo, Por cola o Denegar todo. Al elegir Por cola, especifique una lista de hasta diez colas de origen por nombre de recurso de Amazon (ARN).
8. De forma predeterminada, Amazon SQS proporciona un cifrado del servidor administrado. Para elegir un tipo de clave de cifrado o para desactivar el cifrado del servidor administrado por Amazon SQS, expanda Cifrado. Para obtener más información sobre los tipos de claves de cifrado, consulte Configuración del cifrado del lado del servidor para una cola mediante claves de cifrado administradas por SQL y Configuración del cifrado del lado del servidor para una cola mediante la consola Amazon SQS.

   > **Note**
   >
   > Con SSE activado, se rechazarán las solicitudes `SendMessage` y `ReceiveMessage` anónimas a la cola cifrada. Las prácticas recomendadas de seguridad de Amazon SQS desaconsejan utilizar solicitudes anónimas. Si desea enviar solicitudes anónimas a una cola de Amazon SQS, asegúrese de desactivar SSE.

9. (Opcional) Para configurar una cola de mensajes fallidos a fin de recibir mensajes que no se pueden entregar, expanda Cola de mensajes fallidos.
10. (Opcional) Para agregar etiquetas a la cola, expanda Etiquetas.
11. Elija Crear cola. Amazon SQS crea la cola y muestra la página Detalles de la cola.

Amazon SQS propaga la información sobre la nueva cola por todo el sistema. Dado que Amazon SQS es un sistema distribuido, es posible que experimente un ligero retraso antes de que la consola muestre la cola en la página Colas.

Tras crear una cola, puede enviarle mensajes, así como recibirlos y eliminarlos. También puede editar cualquiera de las opciones de configuración de la cola, excepto el tipo de cola.

Enviar un mensaje

Después de crear la cola, puede enviarle un mensaje.

1. En el panel de navegación izquierdo, elija Colas. En la lista de colas, seleccione la cola que ha creado.
2. En Acciones, elija Enviar y recibir mensajes.

   La consola muestra la página Enviar y recibir mensajes.
3. En Cuerpo del mensaje, especifique el texto del mensaje.
4. Para una cola FIFO (primero en entrar, primero en salir), introduzca un ID de grupo de mensajes. Para obtener más información, consulte Lógica de entrega de colas FIFO en Amazon SQS.
5. (Opcional) Para una cola FIFO, puede introducir un ID de desduplicación de mensajes. Si ha activado la desduplicación basada en el contenido para la cola, el ID de desduplicación de mensajes no es necesario. Para obtener más información, consulte Lógica de entrega de colas FIFO en Amazon SQS.
6. Las colas FIFO no admiten temporizadores en los mensajes individuales. Para obtener más información, consulte Temporizadores de mensajes de Amazon SQS.
7. Elija Enviar mensaje.

   Cuando se envía el mensaje, la consola muestra un mensaje de confirmación. Elija Ver detalles para mostrar información sobre el mensaje enviado.

Administración de una cola de Amazon SQS

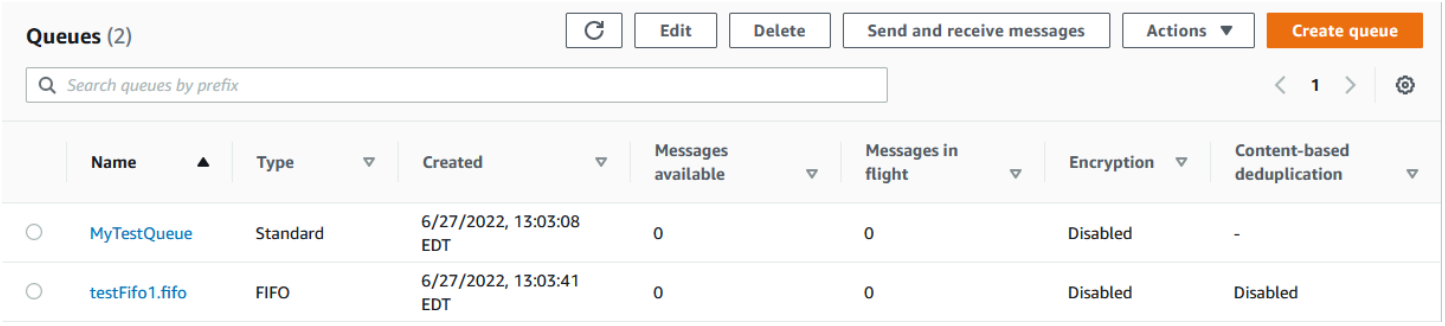
Esta sección lo ayuda a familiarizarse con Amazon SQS al mostrarle cómo administrar las colas y los mensajes con la consola de Amazon SQS.

Requisitos previos

Antes de comenzar, complete los pasos de Configuración de Amazon SQS.

Descripción de la consola de Amazon SQS

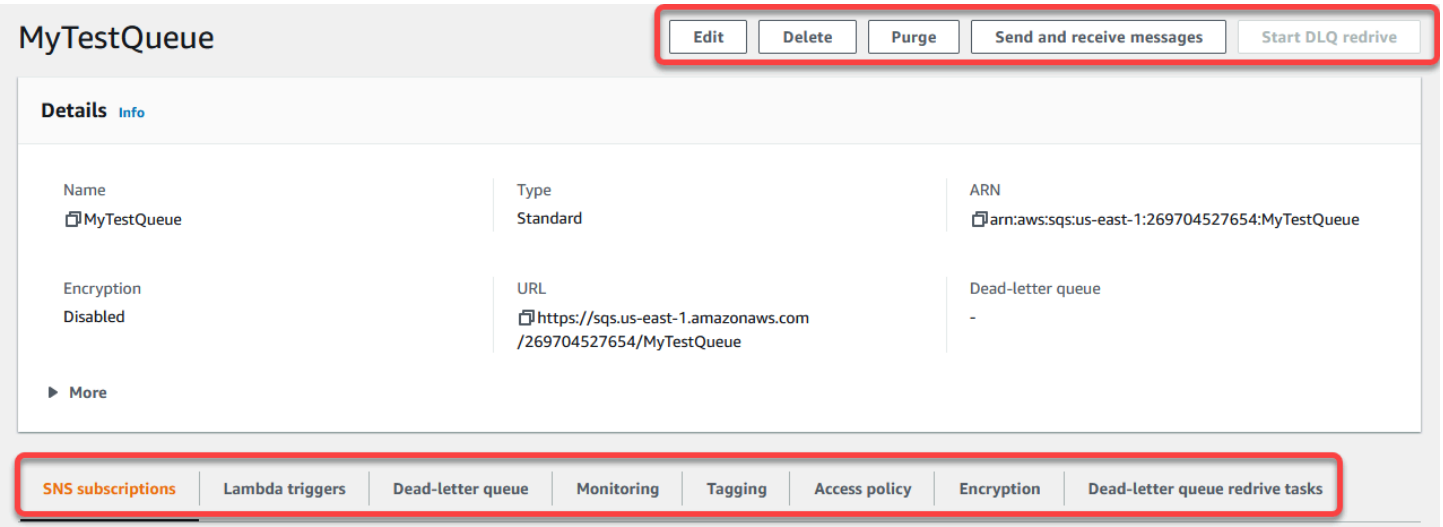
Cuando abra la consola, elija Colas en el panel de navegación para mostrar la página Colas. La página Colas proporciona información sobre todas las colas de la región activa.



La entrada de cada cola muestra el tipo de cola y otra información sobre la cola. La columna Tipo le ayuda a distinguir de un vistazo las colas estándar de las colas FIFO (primero en entrar, primero en salir).

En la página Colas, hay dos formas de realizar acciones en una cola. Puede elegir la opción situada junto al nombre de la cola y, a continuación, elegir la acción que desea realizar en la cola.

También puede elegir el nombre de la cola, que abrirá la página Detalles de la cola. La página Detalles incluye las mismas acciones que la página Colas. Además, puede elegir una de las pestañas situadas debajo de la sección Detalles para ver detalles y acciones de configuración adicionales.



Edición de una cola de Amazon SQS mediante la consola

Puede utilizar la consola Amazon SQS para editar cualquier parámetro de configuración de colas (excepto el tipo de cola) y agregar o eliminar características de cola.

Edición de una cola de Amazon SQS (consola)

1. En la consola de Amazon SQS, abra la página Colas.
2. Seleccione un tema y, a continuación, elija Editar.
3. (Opcional) En Configuración, actualice los parámetros de configuración de la cola.
4. (Opcional) Para actualizar la política de acceso, en Política de acceso, modifique la política JSON.
5. (Opcional) Para actualizar una política de permiso de redireccionamiento de una cola de mensajes fallidos, expanda Política de permiso de redireccionamiento.
6. (Opcional) Para actualizar o eliminar el cifrado, expanda Cifrado.
7. (Opcional) Para agregar, actualizar o eliminar una cola de mensajes fallidos (que le permite recibir mensajes que no se pueden entregar), expanda Cola de mensajes fallidos.
8. (Opcional) Para agregar, actualizar o eliminar las etiquetas de la cola, expanda Etiquetas.
9. Seleccione Guardar.

   La consola muestra la página Detalles de la cola.

Recibir y eliminar un mensaje en Amazon SQS

Tras enviar los mensajes a una cola de Amazon SQS, tiene la opción de recibirlos y eliminarlos. Al solicitar mensajes de una cola, no puede especificar mensajes individuales. En su lugar, usted determina el número máximo de mensajes que desea recuperar, hasta un límite de 10.

Amazon SQS funciona como un sistema distribuido, lo que en ocasiones puede provocar una respuesta vacía al recuperar mensajes de una cola con pocos mensajes. Si esto ocurre, simplemente vuelva a ejecutar la solicitud. Para optimizar la recuperación de mensajes y minimizar las respuestas vacías, considere la posibilidad de utilizar sondeos prolongados. Las votaciones prolongadas retrasan la respuesta hasta que un mensaje esté disponible o se agote el tiempo de espera de la encuesta, lo que reduce los costos innecesarios de las votaciones y mejora la eficiencia.

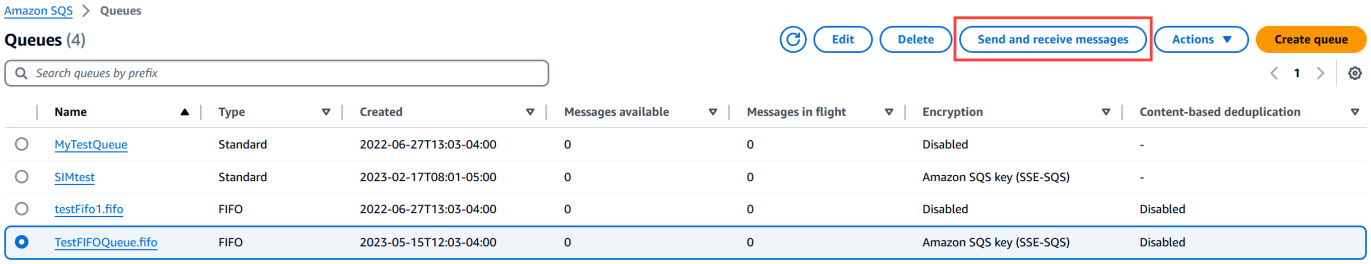
Los mensajes no se eliminan automáticamente tras su recuperación, ya que Amazon SQS garantiza que no pierda el acceso a los mensajes debido a errores de procesamiento, como problemas con la aplicación o interrupciones de la red. Para eliminar permanentemente un mensaje de la cola, debe enviar de forma explícita una solicitud de eliminación después de procesar el mensaje para confirmar que se ha recibido y procesado correctamente.

Cuando los mensajes se recuperan a través de la consola Amazon SQS, se vuelven a ver inmediatamente para volver a recuperarlos. Este comportamiento predeterminado garantiza que los mensajes no se pierdan accidentalmente durante las operaciones manuales, sino que pueden provocar un procesamiento repetido. En entornos automatizados, ajuste la configuración del tiempo de espera de visibilidad para controlar cuánto tiempo un mensaje permanece invisible para otros consumidores después de ser recuperado. Esta configuración es crucial para coordinar el procesamiento de los mensajes entre varios consumidores y garantizar que los mensajes se procesen solo una vez.

Para obtener información más detallada sobre las operaciones de recepción y eliminación de mensajes, consulte la Guía de referencia de la API de Amazon SQS. Esta guía ofrece información completa sobre los puntos de enlace de las API, incluidos los parámetros que permiten gestionar de forma eficaz los escenarios complejos de gestión de mensajes.

Para recibir y eliminar un mensaje mediante la consola

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. En la página Colas, selecciona una cola y, a continuación, selecciona Enviar y recibir mensajes.



4. En la página Enviar y recibir mensajes, selecciona Sondear los mensajes.

   Amazon SQS comienza a sondear los mensajes de la cola. La barra de progreso situada a la derecha de la sección Recibir mensajes muestra la duración del sondeo.

   En la sección Mensajes se muestra una lista de los mensajes recibidos. Para cada mensaje, en la lista se muestran el identificador del mensaje, la fecha de envío, el tamaño y el recuento de recepciones.

5. Para eliminar mensajes, selecciona los mensajes que quieres eliminar y, a continuación, selecciona Eliminar.
6. En el cuadro de diálogo Eliminar mensajes, selecciona Eliminar.

Confirmación de que una cola de Amazon SQS está vacía

En la mayoría de los casos, puede utilizar el sondeo largo para determinar si una cola está vacía. En contados casos, es posible que reciba respuestas vacías incluso cuando una cola aún contenga mensajes, sobre si especifica un valor bajo para Tiempo de espera de recepción del mensaje al crear la cola. En esta sección se describe cómo confirmar que una cola está vacía.

Confirmación de que una cola está vacía (consola)

1. Detenga el envío de mensajes de todos los productores.
2. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
3. En el panel de navegación, elija Queues (Colas).
4. En la página Colas, elija una cola.
5. Elija la pestaña Monitorización.
6. En la parte superior derecha de los paneles de monitorización, elija la flecha hacia abajo situada junto al símbolo Actualizar. En el menú desplegable, elija Actualización automática. Deje el Intervalo de actualización en 1 minuto.

7. Observe los siguientes paneles:

   - Número aproximado de mensajes retrasados
   - Número aproximado de mensajes no visibles
   - Número aproximado de mensajes visibles

   Si todos muestran valores `0` durante varios minutos, la cola está vacía.

Para confirmar que una cola está vacía (AWS CLI, AWS API)

1. Detenga el envío de mensajes de todos los productores.
2. Ejecute uno de los siguientes comandos de forma repetida:

   - AWS CLI: `get-queue-attributes`
   - AWS API: `GetQueueAttributes`
3. Observe las métricas de los siguientes atributos:

   - `ApproximateNumberOfMessagesDelayed`
   - `ApproximateNumberOfMessagesNotVisible`
   - `ApproximateNumberOfMessagesVisible`

   Si todos son `0` durante varios minutos, la cola está vacía.

Si confías en CloudWatch las métricas de Amazon, asegúrate de ver varios puntos de datos cero consecutivos antes de considerar que la cola está vacía. Para obtener más información sobre CloudWatch las métricas, consulta CloudWatch Métricas disponibles para Amazon SQS.

Eliminar una cola de Amazon SQS

Si ya no utiliza una cola de Amazon SQS y no tiene previsto utilizarla en un futuro próximo, le recomendamos que la elimine.

> **Tip**
>
> Si desea verificar que una cola esté vacía antes de eliminarla, consulte Confirmación de que una cola de Amazon SQS está vacía.

Puede eliminar una cola aunque no esté vacía. Para eliminar los mensajes de una cola, pero no la propia cola, púrguela.

Eliminación de una cola (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. En la página Colas, elija la cola que desea eliminar.
4. Elija Eliminar.
5. En el cuadro de diálogo Eliminar cola, introduzca **`delete`** para confirmar la eliminación.
6. Elija Eliminar.

Para eliminar una cola (AWS CLI y una API)

Puede utilizar uno de los siguientes comandos para eliminar una cola:

- AWS CLI: `aws sqs delete-queue`
- AWS API: `DeleteQueue`

Depuración de mensajes de una cola mediante la consola Amazon SQS

Si no desea eliminar una cola de Amazon SQS pero necesita eliminar todos los mensajes que contiene, purgue la cola. El proceso de eliminación de mensajes puede tardar hasta 60 segundos. Le recomendamos que espere 60 segundos independientemente del tamaño de la cola.

> **Important**
>
> Al purgar una cola, no puede recuperar ningún mensaje eliminado.

Purga de una cola (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. En la página Colas, elija la cola que desea purgar.
4. En Acciones, elija Purgar.
5. En el cuadro de diálogo Purgar cola, introduzca **`purge`** y elija Purgar para confirmar el purgado.

   Todos los mensajes se purgan de la cola. La consola mostrará un banner de confirmación.

Tareas habituales para empezar a usar Amazon SQS

Ahora que ha creado una cola y ha aprendido a enviar, recibir y eliminar mensajes y cómo eliminar una cola, es posible que desee probar lo siguiente:

- Para activar una función Lambda, consulte Configuración de una cola de Amazon SQS para activar una función AWS Lambda.
- Aprenda a configurar colas, incluido SSE y otras características.
- Aprenda a enviar un mensaje con atributos.
- Aprenda a enviar un mensaje desde una VPC.
- Para obtener más información sobre la funcionalidad y la arquitectura de Amazon SQS, consulte Tipos de colas de Amazon SQS y Arquitectura de Amazon SQS básica.
- Para obtener instrucciones y advertencias que lo ayudarán a aprovechar al máximo Amazon SQS, consulte Prácticas recomendadas para Amazon SQS.
- Explore los ejemplos de Amazon SQS para uno de los AWS SDK, como la AWS SDK for Java 2.x Guía para desarrolladores.
- Para obtener más información sobre los AWS CLI comandos de Amazon SQS, consulte la Referencia de AWS CLI comandos.
- Para obtener información sobre las acciones de Amazon SQS, consulte la Referencia de la API de Amazon Simple Queue Service.
- Aprenda a interactuar con Amazon SQS mediante programación: lea Cómo trabajar con las API y explore el Centro de desarrollo:AWS
  - Java
  - JavaScript

- PHP
- Python
- Ruby
- Windows y .NET

- Obtenga información sobre cómo controlar los costos y los recursos en la sección Solución de problemas en Amazon SQS.
- Obtenga información sobre la protección de los datos y el acceso a ellos en la sección Seguridad.
- Obtenga más información sobre el flujo de trabajo de Amazon SQS en la sección Flujo de trabajo del proceso de control de acceso de Amazon SQS.

Introducción a las colas estándar de Amazon SQS

Amazon SQS ofrece el tipo estándar como el tipo de cola predeterminado. Las colas estándar admiten un número casi ilimitado de llamadas a la API por segundo, por acción de API (`SendMessage`, `ReceiveMessage` o `DeleteMessage`). Las colas estándar admiten la entrega de at-least-once mensajes. Sin embargo, ocasionalmente (debido a la arquitectura altamente distribuida que permite una capacidad de rendimiento casi ilimitada), puede suceder que más de una copia de un mensaje no se entregue en orden. Las colas estándar proporcionan un protocolo de orden de mejor esfuerzo, lo que garantiza que los mensajes se entreguen en términos generales en el mismo orden en que se envían.

Amazon SQS almacena de forma redundante un mensaje en varias zonas de disponibilidad (AZ) antes de que se confirme un `SendMessage`. Como las copias de los mensajes se almacenan en varias zonas de disponibilidad, ningún error de equipo, red o zona de disponibilidad puede provocar que no se pueda acceder a los mensajes.

Para obtener información sobre cómo crear y configurar colas mediante la consola de Amazon SQS, consulte Cree una cola con la consola Amazon SQS. Para ver ejemplos de Java, consulte Ejemplos de SDK de Java de Amazon SQS.

Puede usar colas de mensajes estándar en muchos escenarios, siempre que su aplicación pueda procesar los mensajes que llegan más de una vez y desordenados, por ejemplo:

- Desacoplar solicitudes de usuario en tiempo real de un trabajo en segundo plano intensivo: permite a los usuarios cargar archivos multimedia mientras cambia el tamaño o se codifican.
- Asignar tareas a varios nodos de empleado: procesa un elevado número de solicitudes de validación de tarjetas de crédito.
- Agrupar mensajes para procesarlos más adelante: programa varias entradas para agregarlas a una base de datos.

Para ver las cuotas relacionadas con las colas estándar, consulte Cuotas.

Para conocer las prácticas recomendadas del trabajo con colas estándar, consulte Recomendaciones para las colas estándar y FIFO de Amazon SQS.

Ordenación de los mensajes

Una cola estándar hace todo lo posible para conservar el orden de los mensajes, pero puede suceder que más de una copia de un mensaje no se entregue en orden. Si el sistema necesita que este orden se conserve, recomendamos utilizar una cola FIFO (primero en entrar, primero en salir) o agregar información de secuenciación en cada mensaje para que pueda reordenar los mensajes cuando se reciban.

Una t-least-once entrega

Amazon SQS almacena copias de los mensajes en varios servidores para mejorar la redundancia y lograr una alta disponibilidad. En raras ocasiones, uno de los servidores que almacena una copia de un mensaje podría no estar disponible cuando usted reciba o elimine un mensaje.

Si esto ocurre, la copia del mensaje no se elimina en el servidor que no está disponible y es posible que vuelva a obtenerla cuando reciba mensajes. Diseñe sus aplicaciones de modo que sean idempotentes (no deben verse afectadas negativamente si se procesa el mismo mensaje más de una vez).

Identificadores de colas y mensajes de Amazon SQS

En esta sección se describen los identificadores de las colas estándar y FIFO. Estos identificadores pueden ayudarle a encontrar y manipular colas y mensajes específicos.

Identificadores de colas estándar de Amazon SQS

Para obtener más información sobre los siguientes identificadores, consulte la Referencia de la API de Amazon Simple Queue Service.

Nombre de cola y URL

Cuando se crea una nueva cola, debe especificar un nombre de cola único para su cuenta y región de AWS . Amazon SQS asigna a cada cola que crea un identificador llamado una URL de cola que incluye el nombre de la cola y otros componentes de Amazon SQS. Siempre que desee realizar una acción en una cola, tiene que proporcionar su URL de cola.

A continuación se muestra la URL de una cola denominada `MyQueue` que pertenece a un usuario que tiene el número de cuenta de AWS `123456789012`.

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue
```

Puede recuperar la dirección URL de una cola mediante programación enumerando las colas y analizando la cadena que sigue al número de cuenta. Para obtener más información, consulte `ListQueues`.

Message ID

Cada mensaje recibe un ID de mensaje asignado por el sistema que Amazon SQS le devuelve en la respuesta `SendMessage`. Este identificador es útil para identificar los mensajes. La longitud máxima de un ID de mensaje es 100 caracteres.

Identificador de recepción

Cada vez que recibe un mensaje de una cola, recibe un identificador de recepción para ese mensaje. Este controlador está asociado a la acción de recepción del mensaje, no al propio mensaje. Para eliminar el mensaje o cambiar la visibilidad de los mensajes, debe proporcionar el identificador de recepción (no el ID de mensaje). Por tanto, siempre debe recibir un mensaje para poder eliminarlo (no puede poner un mensaje en la cola y, a continuación, recuperarlo). La longitud máxima de un identificador de recepción es 1024 caracteres.

> **Important**
>
> Si recibe un mensaje más de una vez, cada vez que lo reciba, obtendrá un identificador de recepción diferente. Cuando solicite eliminar el mensaje, debe proporcionar el identificador de recepción recibido más recientemente (de lo contrario, el mensaje podría no eliminarse).

A continuación se muestra un ejemplo de un identificador de recepción (dividido en tres líneas).

```
MbZj6wDWli+JvwwJaBV+3dcjk2YW2vA3+STFFljTM8tJJg6HRG6PYSasuWXPJB+Cw
Lj1FjgXUv1uSj1gUPAWV66FU/WeR4mq2OKpEGYWbnLmpRCJVAyeMjeU5ZBdtcQ+QE
auMZc8ZRv37sIW2iJKq3M9MFx1YvV11A2x/KSbkJ0=
```

Cuotas

En la siguiente tabla se muestran las cuotas relacionadas con las colas estándar.

Cuota	Descripción
Cola con retraso	El retraso predeterminado (mínimo) de una cola es de 0 segundos. El valor máximo es de 15 minutos.
Colas mostradas	1000 colas por cada solicitud de `ListQueues` .
Tiempo de espera de sondeo largo	El tiempo máximo de espera de sondeo es de 20 segundos.
Mensajes por cola (pendientes)	El número de mensajes que puede almacenar una cola de Amazon SQS es ilimitado.
Mensajes por cola (en tránsito)	En la mayoría de las colas estándar (según el tráfico de colas y los mensajes atrasados), puede haber un máximo de aproximadamente 120 000 mensajes en tránsito (recibidos de una cola por un consumidor, pero que aún no se han eliminado de la cola). Si alcanza esta cuota mientras utiliza sondeos cortos, Amazon SQS devuelve el mensaje de error `OverLimit` . Si utiliza sondeos largos, Amazon SQS no devuelve ningún mensaje de error. Para evitar llegar a la cuota, conviene eliminar los mensajes de la cola una vez procesados. También puede aumentar el número de las colas que usa para procesar los mensajes. Para solicitar un aumento de la cuota, envíe una solicitud de soporte técnico.
Nombre de la cola	El nombre de una cola puede tener hasta 80 caracteres. Se aceptan los siguientes caracteres: caracteres alfanuméricos, guiones (`-`) y guiones bajos (`_`). **Note** Los nombres de cola distinguen entre mayúsculas y minúsculas (por ejemplo, `Test-queue` y `test-queue` son colas diferentes).

Cuota	Descripción
Etiqueta de la cola	No recomendamos agregar más de 50 etiquetas a una cola. El etiquetado admite caracteres Unicode en UTF-8.
	Se requiere la etiqueta `Key`, pero la etiqueta `Value` es opcional.
	La etiqueta `Key` y la etiqueta `Value` distinguen entre mayúsculas y minúsculas.
	La pestaña `Key` y la pestaña `Value` pueden incluir caracteres alfanuméricos en UTF-8 y espacios en blanco. Se permiten los siguientes caracteres especiales: `_ . : / = + - @`
	La etiqueta `Key` o `Value` no debe incluir el prefijo reservado `aws:` (con este prefijo, no puede eliminar las claves o los valores de una etiqueta).
	La longitud máxima de la etiqueta `Key` es de 128 caracteres Unicode en UTF-8. La etiqueta `Key` no debe estar vacía o ser nula.
	La longitud máxima de la etiqueta `Value` es de 256 caracteres Unicode en UTF-8. La etiqueta `Value` puede estar vacía o ser nula.
	Las acciones de etiquetado están limitadas a 30 TPS cada una. Cuenta de AWS Si su aplicación requiere un mayor rendimiento, envíe una solicitud.

Introducción a las colas FIFO en Amazon SQS

Las colas FIFO (primero en entrar, primero en salir) tienen todas las funciones de las colas estándar, pero están diseñadas para mejorar los mensajes entre aplicaciones cuando el orden de las operaciones y los eventos es fundamental, o cuando no se puede tolerar la existencia de duplicados.

Algunos ejemplos de situaciones en las que puede utilizar colas FIFO son las siguientes:

1. Sistema de administración de pedidos de comercio electrónico en el que el orden es fundamental
2. Integración con sistemas de terceros en los que es necesario procesar eventos en orden
3. Procesamiento de las entradas introducidas por el usuario en el orden introducido
4. Comunicaciones y redes: envío y recepción de datos e información en el mismo orden
5. Sistemas de computación: comprobación de que los comandos introducidos por el usuario se ejecutan en el orden correcto
6. Institutos de enseñanza: evitar que un estudiante se matricule en un curso antes de registrarse para obtener una cuenta
7. Sistema de venta de entradas en línea: las entradas se distribuyen por orden de llegada

> **Note**
>
> Las colas FIFO también proporcionan un procesamiento único, pero tienen un número limitado de transacciones por segundo (TPS). Puede utilizar el modo de alto rendimiento de Amazon SQS con la cola FIFO para aumentar el límite de transacciones. Para obtener más información sobre el uso del modo de alto rendimiento, consulte Alto rendimiento de las colas FIFO en Amazon SQS. Para obtener información sobre las cuotas de rendimiento, consulte the section called “Cuotas de mensajes”.

Las colas FIFO de Amazon SQS están disponibles en todas las regiones en las que está disponible Amazon SQS.

Para obtener más información sobre cómo utilizar las colas FIFO con ordenaciones complejas, consulte Resolución de problemas de ordenaciones complejas con las colas FIFO de Amazon SQS.

Para obtener información sobre cómo crear y configurar colas mediante la consola de Amazon SQS, consulte Cree una cola con la consola Amazon SQS. Para ver ejemplos de Java, consulte Ejemplos de SDK de Java de Amazon SQS.

Para conocer las prácticas recomendadas del trabajo con colas FIFO, consulte Recomendaciones adicionales para las colas FIFO de Amazon SQS y Recomendaciones para las colas estándar y FIFO de Amazon SQS.

Lógica de entrega de colas FIFO en Amazon SQS

Los siguientes conceptos puede ayudarlo a entender mejor cómo se envían y reciben mensajes desde FIFO.

Envío de mensajes

Si se envían varios mensajes de forma sucesiva a una cola FIFO, cada uno de ellos con un ID de desduplicación de mensajes distinto, Amazon SQS almacena los mensajes y reconoce la transmisión. A continuación, se puede recibir y procesar cada mensaje en el mismo orden en que se transmitieron.

En las colas FIFO, los mensajes se ordenan en función del ID de grupo de mensajes. Si varios hosts (o diferentes subprocesos en el mismo host) envían mensajes con el mismo ID de grupo de mensajes a una cola FIFO, Amazon SQS almacena los mensajes en el orden en que llegan para su procesamiento. Para asegurarse de que Amazon SQS conserva el orden de envío y recepción de los mensajes, cada productor debe utilizar un ID de grupo de mensajes único para enviar todos sus mensajes.

La lógica de las colas FIFO se aplica únicamente por ID de grupo de mensajes. Cada ID de grupo de mensajes representa un grupo de mensajes con una ordenación distinta dentro de una cola de Amazon SQS. Para cada ID de grupo de mensajes, todos los mensajes se envían y reciben por riguroso orden. No obstante, es posible que los mensajes con diferentes valores de ID de grupo de mensajes no se envíen y reciban en orden. Debe asociar un ID de grupo de mensajes al mensaje. Si no proporciona un ID de grupo de mensajes, la acción da error. Si necesita un único grupo de mensajes ordenados, proporcione el mismo ID de grupo de mensajes para los mensajes enviados a la cola FIFO.

Recepción de mensajes

No puede solicitar la recepción de mensajes con un ID de grupo de mensajes específico.

Cuando se reciben mensajes de una cola FIFO con varios ID de grupo de mensajes, Amazon SQS intenta primero devolver el máximo número de mensajes que tienen el mismo ID de grupo de mensajes como sea posible. Esto permite que otros consumidores procesen los mensajes que tienen un ID de grupo de mensajes diferente. Cuando reciba un mensaje con un ID de grupo de mensajes, no se devolverán más mensajes para el mismo ID de grupo de mensajes a menos que elimine el mensaje o este se haga visible.

> **Note**
>
> Es posible recibir hasta 10 mensajes en una sola llamada mediante el parámetro de solicitud `MaxNumberOfMessages` de la acción `ReceiveMessage` de la Estos mensajes retienen su orden FIFO y pueden tener el mismo ID de grupo de mensajes. Por lo tanto, si hay menos de 10 mensajes disponibles con el mismo ID de grupo de mensajes, es posible que reciba mensajes de otro ID de grupo de mensajes en el mismo lote de diez mensajes, pero aun así en orden FIFO.

Varios reintentos

Las colas FIFO permiten al productor o al consumidor efectuar múltiples reintentos:

- Si el productor detecta una acción `SendMessage` con error, puede reintentar el envío tantas veces como sea necesario, mediante el mismo ID de desduplicación de mensajes. Si se supone que el productor recibe al menos un acuse de recibo antes de que caduque el intervalo de desduplicación, los múltiples reintentos no afectan al orden de los mensajes ni generan duplicados.
- Si el consumidor detecta una acción `ReceiveMessage` con error, puede volver a intentarlo tantas veces como sea necesario, mediante el mismo ID de intento de solicitud de recepción. Si se supone que el consumidor recibe al menos un acuse de recibo antes de que caduque el tiempo de visibilidad, los múltiples reintentos no afectan al orden de los mensajes.
- Cuando reciba un mensaje con un ID de grupo de mensajes, no se devolverán más mensajes para el mismo ID de grupo de mensajes a menos que elimine el mensaje o este se haga visible.

Ordenación de mensajes en cola FIFO en Amazon SQS

La cola FIFO mejora y complementa la cola estándar. Las características más importantes de este tipo de cola son entrega FIFO (primero en entrar, primero en salir) y procesamiento único:

- El orden en el que se envían y reciben los mensajes se conserva estrictamente, y los mensajes se entregan una vez y no están disponibles hasta que un consumidor los procese y los elimine.
- No se introducen duplicados en la cola.

Además, las colas FIFO admiten grupos de mensajes que permiten varios grupos de mensajes ordenados en una única cola. No hay cuota para el número de grupos de mensajes en una cola FIFO.

Procesamiento de una sola vez en Amazon SQS

A diferencia de las colas estándar, las colas FIFO no generan mensajes duplicados. Las colas FIFO le ayudan a evitar el envío de duplicados a una cola. Si reintenta la acción `SendMessage` dentro del intervalo de desduplicación de cinco minutos, Amazon SQS no genera ningún duplicado en la cola.

Para configurar la desduplicación, debe realizar una de las siguientes acciones:

- Habilitar la desduplicación basada en el contenido. Esto indica a Amazon SQS que debe utilizar un hash SHA-256 para generar el ID de desduplicación de mensajes mediante el cuerpo del mensaje, pero no los atributos del mensaje. Para obtener más información, consulte la documentación de las acciones `CreateQueue`, `GetQueueAttributes` y `SetQueueAttributes` en la Referencia de la API de Amazon Simple Queue Service.
- Proporcionar de manera explícita el ID de desduplicación de mensajes (o ver el número secuencial) del mensaje. Para obtener más información, consulte la documentación de las acciones `SendMessage`, `SendMessageBatch` y `ReceiveMessage` en la Referencia de la API de Amazon Simple Queue Service.

Pasar de una cola estándar a una cola FIFO en Amazon SQS

Si dispone de una aplicación que utiliza colas estándar y desea aprovechar las características de ordenación o procesamiento único de las colas FIFO, debe configurar correctamente tanto la cola como la aplicación.

> **Note**
>
> No puede convertir una cola estándar existente en una cola FIFO. Para realizar el cambio, debe crear una nueva cola FIFO para su aplicación o eliminar su cola estándar existente y volver a crearla como una cola FIFO.

Para asegurarse de que su aplicación funciona correctamente con una cola FIFO, utilice la siguiente lista de comprobación:

- Utilice el modo de alto rendimiento recomendado para FIFO a fin de lograr un mayor rendimiento. Para obtener más información acerca de las cuotas de mensajes, consulte Cuotas de mensajes de Amazon SQS.
- Las colas FIFO no admiten retrasos por mensaje, solo por cola. Si su aplicación establece el mismo valor del parámetro `DelaySeconds` en todos los mensajes, debe modificar la aplicación para eliminar el retraso por mensaje y establecer `DelaySeconds` en toda la cola.
- El grupo de mensajes es una característica FIFO única que permite a los clientes procesar mensajes en paralelo a la vez que mantienen su orden respectivo. Los clientes organizan los mensajes en grupos de mensajes mediante la especificación de un ID de grupo de mensajes. Los grupos de mensajes suelen basarse en una dimensión empresarial para una carga de trabajo determinada. Para escalar mejor con colas FIFO, utilice una dimensión empresarial más detallada para el ID de mensaje. Cuantos más identificadores de grupos de mensajes distribuya, mayor será el número de mensajes que FIFO ponga a disposición para su consumo.
- Antes de enviar mensajes a una cola FIFO, confirme lo siguiente:
  - Si su aplicación puede enviar mensajes con cuerpos idénticos, puede modificar la aplicación para proporcionar un ID único de desduplicación de mensajes para cada mensaje enviado.
  - Si su aplicación envía mensajes con cuerpos únicos, puede habilitar la desduplicación basada en el contenido.
- No es necesario hacer ningún cambio en el código del consumidor. Sin embargo, si se tarda mucho tiempo en procesar los mensajes y el tiempo de espera de visibilidad es un alto valor, considere la posibilidad de añadir un ID de intento de solicitud de recepción para cada acción `ReceiveMessage`. Esto le permite reintentar la recepción en caso de que se produzcan errores de red e impide que las colas se pausen debido a intentos fallidos de recepción.

Para obtener más información, consulte la Referencia de la API de Amazon Simple Storage Service.

Alto rendimiento de las colas FIFO en Amazon SQS

Las colas FIFO de alto rendimiento de Amazon SQS gestionan de forma eficiente el alto rendimiento de los mensajes y, al mismo tiempo, mantienen un orden estricto de los mensajes, lo que garantiza la fiabilidad y la escalabilidad de las aplicaciones que procesan numerosos mensajes. Esta solución es ideal para situaciones que exigen tanto un alto rendimiento como una entrega de mensajes ordenada.

Las colas FIFO de alto rendimiento de Amazon SQS no son necesarias en situaciones en las que el orden estricto de los mensajes no es crucial y en las que el volumen de mensajes entrantes es relativamente bajo o esporádico. Por ejemplo, si tiene una aplicación a pequeña escala que procesa mensajes poco frecuentes o no secuenciales, es posible que la complejidad y el costo adicionales asociados a las colas FIFO de alto rendimiento no estén justificados. Además, si su aplicación no requiere las capacidades de rendimiento mejoradas que ofrecen las colas FIFO de alto rendimiento, optar por una cola Amazon SQS estándar podría resultar más rentable y sencillo de gestionar.

Para mejorar la capacidad de solicitudes en las colas FIFO de alto rendimiento, se recomienda aumentar el número de grupos de mensajes. Para obtener más información sobre las cuotas de mensajes de alto rendimiento, consulte Service Quotas de Amazon SQS en la Referencia general de Amazon Web Services.

Para obtener información sobre las cuotas por cola y las estrategias de distribución de datos, consulte y. Cuotas de mensajes de Amazon SQS Particiones y distribución de datos para obtener alto rendimiento en las colas FIFO de SQS

Temas

- Casos de uso de alto rendimiento para colas FIFO de Amazon SQS
- Particiones y distribución de datos para obtener alto rendimiento en las colas FIFO de SQS
- Permita un alto rendimiento para las colas FIFO en Amazon SQS

Casos de uso de alto rendimiento para colas FIFO de Amazon SQS

Los siguientes casos de uso destacan las diversas aplicaciones de las colas FIFO de alto rendimiento y muestran su eficacia en todos los sectores y escenarios:

1. Procesamiento de datos en tiempo real: las aplicaciones que se ocupan de flujos de datos en tiempo real, como el procesamiento de eventos o la ingesta de datos telemétricos, pueden

beneficiarse de las colas FIFO de alto rendimiento para gestionar la afluencia continua de mensajes y, al mismo tiempo, conservar su orden para un análisis preciso.

2. Procesamiento de pedidos de comercio electrónico: en las plataformas de comercio electrónico en las que es fundamental mantener el orden de las transacciones de los clientes, las colas FIFO de alto rendimiento garantizan que los pedidos se procesen de forma secuencial y sin demoras, incluso durante las temporadas altas de compras.
3. Servicios financieros: las instituciones financieras que gestionan datos comerciales o transaccionales de alta frecuencia utilizan colas FIFO de alto rendimiento para procesar los datos de mercado y las transacciones con una latencia mínima y, al mismo tiempo, cumplen con los estrictos requisitos reglamentarios para la gestión de mensajes.
4. Transmisión multimedia: las plataformas de transmisión y los servicios de distribución multimedia utilizan colas FIFO de alto rendimiento para gestionar la entrega de archivos multimedia y contenido en streaming, lo que garantiza una experiencia de reproducción fluida para los usuarios y, al mismo tiempo, mantiene el orden correcto de entrega del contenido.

Particiones y distribución de datos para obtener alto rendimiento en las colas FIFO de SQS

Amazon SQS almacena los datos de la cola FIFO en particiones. Una partición es una asignación de almacenamiento para una cola que se replica automáticamente en varias zonas de disponibilidad de una región. AWS Usted no administra las particiones. En su lugar, Amazon SQS se encarga de la administración de particiones.

Para las colas FIFO, Amazon SQS modifica el número de particiones de una cola en las siguientes situaciones:

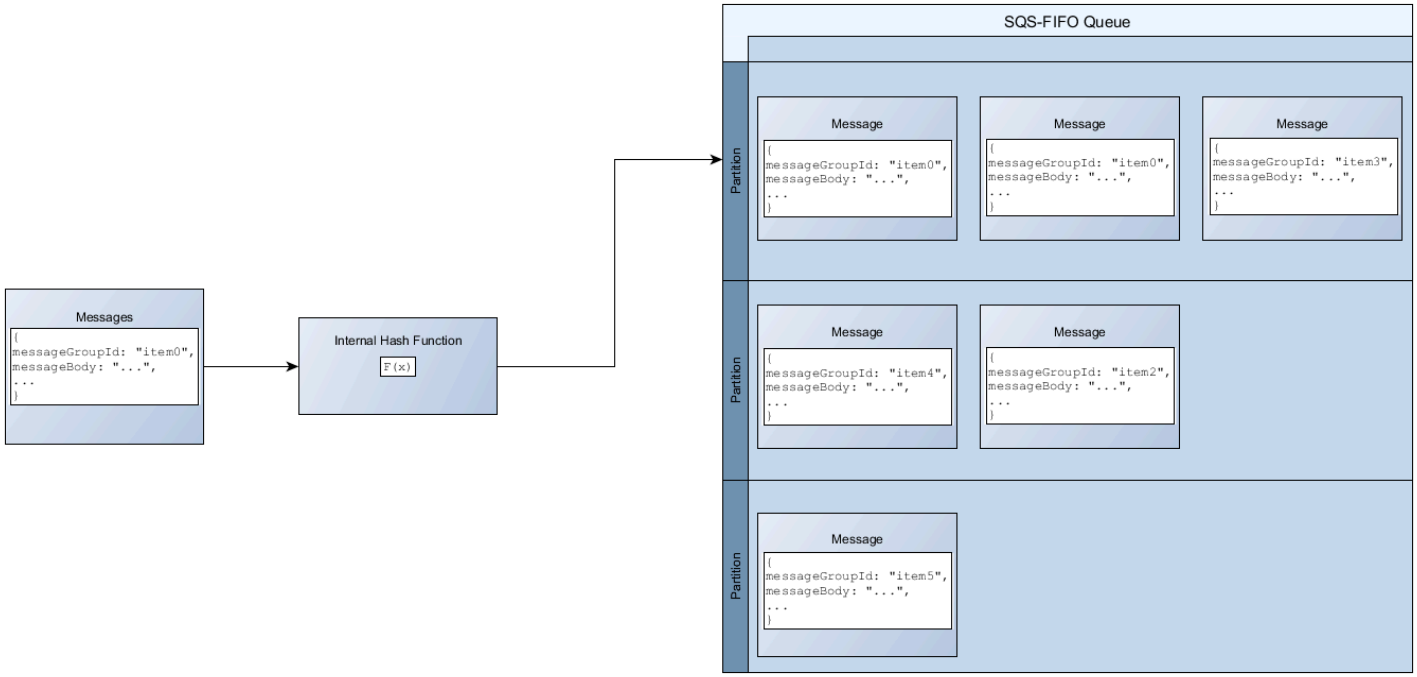
- Si la tasa de solicitudes actual se aproxima o supera lo que pueden admitir las particiones existentes, se asignan particiones adicionales hasta que la cola alcanza la cuota regional. Para obtener información sobre las cuotas, consulte Cuotas de mensajes de Amazon SQS.
- Si las particiones actuales se utilizan poco, es posible que se reduzca el número de particiones.

La administración de las particiones tiene lugar automáticamente en segundo plano y es transparente para las aplicaciones. Su cola y sus mensajes están disponibles en todo momento.

Distribución de los datos por ID de grupo de mensajes

Para agregar un mensaje a una cola FIFO, Amazon SQS utiliza el valor del ID de grupo de mensajes de cada mensaje como entrada a una función hash interna. El valor del resultado de la función hash determina la partición que almacena el mensaje.

En el siguiente diagrama se muestra una tabla denominada que abarca varias particiones. El ID de grupo de mensajes de la cola se basa en el número de elemento. Amazon SQS utiliza la función hash para determinar dónde se almacenará un nuevo elemento; en este caso, se basa en el valor hash de la cadena `item0`. Tenga en cuenta que los elementos se almacenan en el mismo orden en el que se agregan a la cola. La ubicación de cada elemento viene determinada por el valor hash de su ID de grupo de mensajes.



> **Note**
>
> Amazon SQS está optimizado para una distribución uniforme de los elementos en las particiones de una cola FIFO, independientemente del número de particiones. AWS recomienda utilizar identificadores de grupos de mensajes que puedan tener un gran número de valores distintos.

Optimización de la utilización de particiones

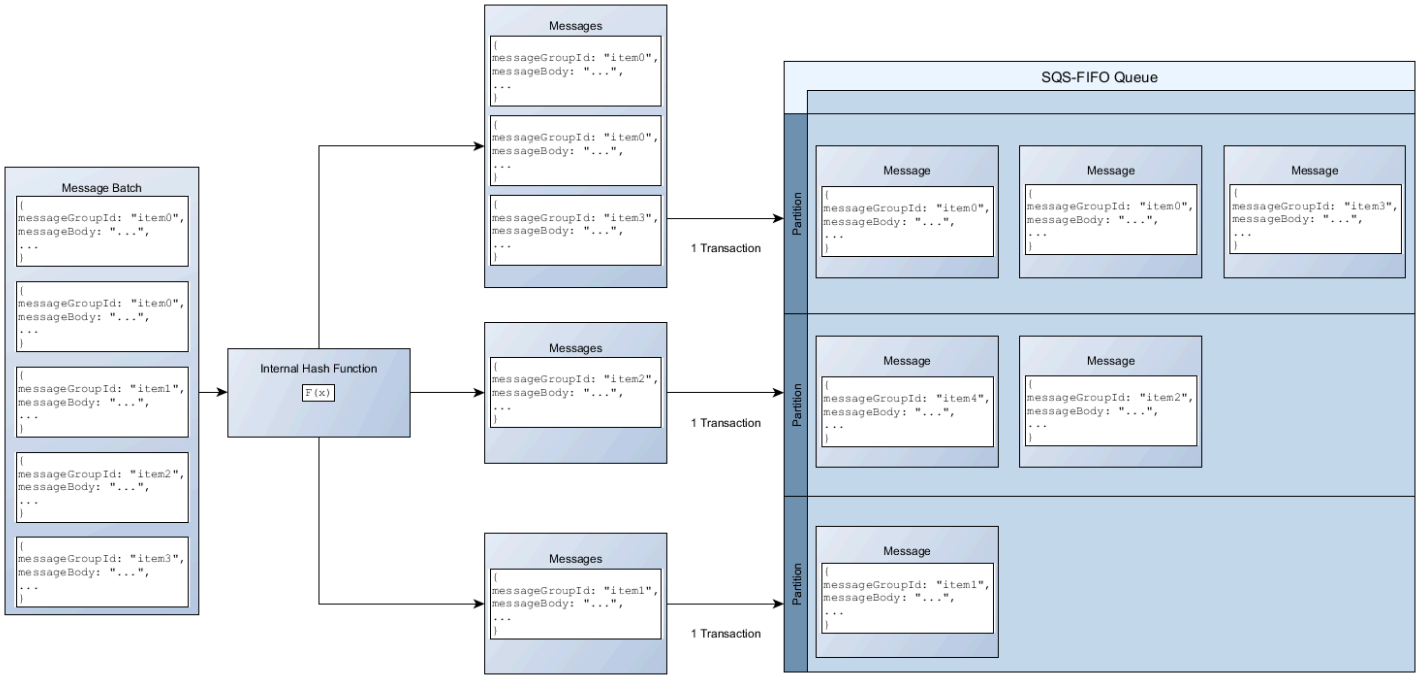
Cada partición admite hasta 3000 mensajes por segundo con procesamiento por lotes o hasta 300 mensajes por segundo para operaciones de envío, recepción y eliminación en las regiones compatibles. Para obtener más información sobre las cuotas de mensajes de alto rendimiento, consulte Service Quotas de Amazon SQS en la Referencia general de Amazon Web Services.

Cuando se utilizan las API por lotes, cada mensaje se enruta según el proceso descrito en Distribución de los datos por ID de grupo de mensajes. Los mensajes que se enrutan a la misma partición se agrupan y procesan en una única transacción.

Para optimizar el uso de las particiones de la `SendMessageBatch` API, AWS recomienda agrupar los mensajes por lotes con los mismos ID de grupo de mensajes siempre que sea posible.

Para optimizar el uso de las particiones en `ChangeMessageVisibilityBatch` las API `DeleteMessageBatch` y en las API, se AWS recomienda utilizar `ReceiveMessage` las solicitudes con el `MaxNumberOfMessages` parámetro establecido en 10 y agrupar en lotes los identificadores de recepción devueltos por una sola solicitud. `ReceiveMessage`

En el siguiente ejemplo, se envía un lote de mensajes con varios ID de grupo de mensajes. El lote se divide en tres grupos, cada uno de los cuales cuenta para la cuota de la partición.



> **Note**
>
> Amazon SQS solo garantiza que los mensajes con la misma función hash interna de ID de grupo de mensajes se agrupen en una solicitud por lotes. En función del resultado de la función hash interna y del número de particiones, se podrían agrupar los mensajes con ID de grupo de mensajes diferentes. Dado que la función hash o el número de particiones pueden cambiar en cualquier momento, los mensajes agrupados en un momento dado pueden no estarlo más adelante.

Permita un alto rendimiento para las colas FIFO en Amazon SQS

Puede habilitar el alto rendimiento para cualquier cola FIFO nueva o existente. Esta característica incluye tres nuevas opciones a la hora de crear y editar colas FIFO:

- Habilitar FIFO de alto rendimiento: aumenta el rendimiento de los mensajes en la cola FIFO actual.
- Ámbito de desduplicación: Especifica si la desduplicación se produce en el nivel de cola o de grupo de mensajes..
- Límite de rendimiento FIFO: especifica si la cuota de rendimiento de los mensajes en la cola FIFO se establece en el nivel de cola o de grupo de mensajes..

Habilitación de un alto rendimiento en una cola FIFO (consola)

1. Inicie la creación o edición de una cola FIFO.
2. Cuando especifique las opciones para la cola, elija Habilitar FIFO de alto rendimiento.

   La activación del alto rendimiento para las colas FIFO establece las opciones relacionadas de la siguiente manera:

   - Ámbito de desduplicación se establece a Grupo de mensajes, la configuración necesaria para utilizar alto rendimiento en las colas FIFO.
   - Límite de rendimiento FIFO se establece a Por ID de grupo de mensajes, la configuración necesaria para utilizar alto rendimiento en las colas FIFO.

   Si cambia alguna de las configuraciones necesarias para utilizar colas FIFO de alto rendimiento, el rendimiento normal estará en vigor para la cola y la desduplicación se producirá según lo especificado.

3. Continúe con la especificación de todas las opciones para la cola. Cuando termine, elija Crear cola o Guardar.

Tras crear o editar la cola FIFO, puede enviar mensajes a ella y recibir y eliminar mensajes, todo ello en un TPS superior. Para obtener información de las cuotas de alto rendimiento, consulte Rendimiento de mensajes en Cuotas de mensajes de Amazon SQS.

Términos clave de Amazon SQS

Los siguientes términos clave pueden ayudarlo a comprender mejor la funcionalidad de las colas FIFO. Para obtener más información, consulte la Referencia de la API de Amazon Simple Queue Service.

ID de desduplicación de mensajes

El token utilizado para la desduplicación de los mensajes enviados. Si un mensaje con un ID de desduplicación de mensajes concreto se envía correctamente, todos los mensajes enviados con el mismo ID de desduplicación de mensajes se aceptan correctamente pero no se entregan durante el intervalo de desduplicación de 5 minutos.

> **Note**
>
> Amazon SQS sigue realizando un seguimiento del ID de desduplicación del mensaje incluso después de haberlo recibido y eliminado.

ID de grupo de mensajes

La etiqueta que especifica que un mensaje pertenece a un grupo de mensajes específico. Los mensajes que pertenecen al mismo grupo de mensajes se procesan siempre uno a uno, en un orden estricto relativo al grupo de mensajes (no obstante, los mensajes que pertenecen a grupos de mensajes diferentes podrían procesarse sin orden).

ID de intento de solicitud de recepción

El token que se utiliza para la desduplicación de llamadas a `ReceiveMessage`.

Número de secuencia

El número grande y no consecutivo que Amazon SQS asigna a cada mensaje.

Compatibilidad con FIFO en Amazon SQS

Clientes

El cliente asincrónico con búfer de Amazon SQS no admite actualmente las colas FIFO.

Servicios

Si su aplicación utiliza varios AWS servicios o una combinación de AWS servicios externos, es importante entender qué funcionalidad de servicio no admite las colas FIFO.

Es posible que algunos servicios externos AWS o algunos que envían notificaciones a Amazon SQS no sean compatibles con las colas FIFO, a pesar de que le permiten establecer una cola FIFO como destino.

Las siguientes características de los AWS servicios no son compatibles actualmente con las colas FIFO:

- Notificaciones de eventos de Amazon S3
- Enlaces de ciclo de vida de escalado automático
- AWS IoT Acciones de reglas
- AWS Lambda Colas de mensajes fallidos

Para obtener información acerca de la compatibilidad de otros servicios con colas FIFO, consulte la documentación de los servicios.

Identificadores de colas y mensajes FIFO en Amazon SQS

En esta sección se describen los identificadores de las colas FIFO. Estos identificadores pueden ayudarle a encontrar y manipular colas y mensajes específicos.

Temas

- Identificadores de colas FIFO en Amazon SQS
- Identificadores adicionales para las colas FIFO de Amazon SQS

Identificadores de colas FIFO en Amazon SQS

Para obtener más información sobre los siguientes identificadores, consulte la Referencia de la API de Amazon Simple Queue Service.

Nombre de cola y URL

Cuando se crea una nueva cola, debe especificar un nombre de cola único para su cuenta y región de AWS . Amazon SQS asigna a cada cola que crea un identificador llamado una URL de cola que incluye el nombre de la cola y otros componentes de Amazon SQS. Siempre que desee realizar una acción en una cola, tiene que proporcionar su URL de cola.

La cola FIFO debe finalizar con el sufijo `.fifo`. El sufijo cuenta para la cuota de nombre de cola de 80 caracteres. Para determinar si una cola es FIFO, puede comprobar si el nombre de la cola termina con el sufijo.

La siguiente es la URL de cola de una cola FIFO denominada `MyQueue` propiedad de un usuario con el número de cuenta de AWS. `123456789012`

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue.fifo
```

Puede recuperar la dirección URL de una cola mediante programación enumerando las colas y analizando la cadena que sigue al número de cuenta. Para obtener más información, consulte `ListQueues`.

Message ID

Cada mensaje recibe un ID de mensaje asignado por el sistema que Amazon SQS le devuelve en la respuesta `SendMessage`. Este identificador es útil para identificar los mensajes. La longitud máxima de un ID de mensaje es 100 caracteres.

Identificador de recepción

Cada vez que recibe un mensaje de una cola, recibe un identificador de recepción para ese mensaje. Este controlador está asociado a la acción de recepción del mensaje, no al propio mensaje. Para eliminar el mensaje o cambiar la visibilidad de los mensajes, debe proporcionar el identificador de recepción (no el ID de mensaje). Por tanto, siempre debe recibir un mensaje para poder eliminarlo (no puede poner un mensaje en la cola y, a continuación, recuperarlo). La longitud máxima de un identificador de recepción es 1024 caracteres.

> **Important**
>
> Si recibe un mensaje más de una vez, cada vez que lo reciba, obtendrá un identificador de recepción diferente. Cuando solicite eliminar el mensaje, debe proporcionar el identificador de recepción recibido más recientemente (de lo contrario, el mensaje podría no eliminarse).

A continuación se muestra un ejemplo de un identificador de recepción (dividido en tres líneas).

```
MbZj6wDWli+JvwwJaBV+3dcjk2YW2vA3+STFFljTM8tJJg6HRG6PYSasuWXPJB+Cw
Lj1FjgXUv1uSj1gUPAWV66FU/WeR4mq2OKpEGYWbnLmpRCJVAyeMjeU5ZBdtcQ+QE
auMZc8ZRv37sIW2iJKq3M9MFx1YvV11A2x/KSbkJ0=
```

Identificadores adicionales para las colas FIFO de Amazon SQS

Para obtener más información sobre los siguientes identificadores, consulte Procesamiento de una sola vez en Amazon SQS y la Referencia de la API de Amazon Simple Queue Service.

ID de desduplicación de mensajes

El token utilizado para la desduplicación de los mensajes enviados. Si un mensaje con un ID de desduplicación de mensajes concreto se envía correctamente, todos los mensajes enviados con el mismo ID de desduplicación de mensajes se aceptan correctamente pero no se entregan durante el intervalo de desduplicación de 5 minutos.

ID de grupo de mensajes

La etiqueta que especifica que un mensaje pertenece a un grupo de mensajes específico. Los mensajes que pertenecen al mismo grupo de mensajes se procesan siempre uno a uno, en un orden estricto relativo al grupo de mensajes (no obstante, los mensajes que pertenecen a grupos de mensajes diferentes podrían procesarse sin orden).

Número de secuencia

El número grande y no consecutivo que Amazon SQS asigna a cada mensaje.

Cuotas de Amazon SQS

En este tema se enumeran cuotas de Amazon Simple Queue Service (Amazon SQS).

Temas

Cuotas de colas FIFO de Amazon SQS

Cuotas de Amazon SQS

En la siguiente tabla se muestran las cuotas relacionadas con las colas FIFO.

Cuota	Descripción
Cola con retraso	El retraso predeterminado (mínimo) de una cola es de 0 segundos. El valor máximo es de 15 minutos.
Colas mostradas	1000 colas por cada solicitud de `ListQueues` .
Tiempo de espera de sondeo largo	El tiempo máximo de espera de sondeo es de 20 segundos.
Grupos de mensajes	No hay cuota para el número de grupos de mensajes en una cola FIFO.
Mensajes por cola (pendientes)	El número de mensajes que puede almacenar una cola de Amazon SQS es ilimitado.
Mensajes por cola (en tránsito)	En el caso de las colas FIFO, puede haber un máximo de 20 000 mensajes en tránsito (recibidos de una cola por un consumidor, pero aún no eliminados de la cola). Si alcanza esta cuota, Amazon SQS no devuelve ningún mensaje de error.

Cuota	Descripción
Nombre de la cola	La cola FIFO debe finalizar con el sufijo `.fifo`. El sufijo cuenta para la cuota de nombre de cola de 80 caracteres. Para determinar si una cola es FIFO, puede comprobar si el nombre de la cola termina con el sufijo.
Etiqueta de la cola	No recomendamos agregar más de 50 etiquetas a una cola. El etiquetado admite caracteres Unicode en UTF-8.
	Se requiere la etiqueta `Key`, pero la etiqueta `Value` es opcional.
	La etiqueta `Key` y la etiqueta `Value` distinguen entre mayúsculas y minúsculas.
	La pestaña `Key` y la pestaña `Value` pueden incluir caracteres alfanuméricos en UTF-8 y espacios en blanco. Se permiten los siguientes caracteres especiales: `_ . : / = + - @`
	La etiqueta `Key` o `Value` no debe incluir el prefijo reservado `aws:` (con este prefijo, no puede eliminar las claves o los valores de una etiqueta).
	La longitud máxima de la etiqueta `Key` es de 128 caracteres Unicode en UTF-8. La etiqueta `Key` no debe estar vacía o ser nula.
	La longitud máxima de la etiqueta `Value` es de 256 caracteres Unicode en UTF-8. La etiqueta `Value` puede estar vacía o ser nula.
	Las acciones de etiquetado están limitadas a 30 TPS por cada una. Cuenta de AWS Si su aplicación requiere un mayor rendimiento, envíe una solicitud.

Cuotas de mensajes de Amazon SQS

En la siguiente tabla se muestran las cuotas relacionadas con los mensajes.

Cuota	Descripción
ID de mensaje por lotes	Un ID de mensaje por lotes puede tener hasta 80 caracteres. Se aceptan los siguientes caracteres: caracteres alfanuméricos, guiones (-) y guiones bajos (_).
Atributos de mensajes	Un mensaje puede contener hasta 10 atributos de metadatos.
Lote de mensajes	Una única solicitud por lotes de mensajes puede incluir un máximo de 10 mensajes. Para obtener más informaci ón, consulte Configuración del cliente AmazonSQS BufferedAsync en la sección Acciones por lotes de Amazon SQS.
Contenido de los mensajes	Un mensaje solo puede incluir XML, JSON y texto sin formato. Se permiten los siguientes caracteres Unicode: `#x9` | `#xA` | `#xD` | `#x20` a `#xD7FF` | `#xE000` a `#xFFFD` | `#x10000` a `#x10FFFF` Cualquier carácter que no esté incluido en esta lista se rechazará. Para obtener más información, consulte la especificación W3C respecto a los caracteres.
ID de grupo de mensajes	Consuma los mensajes de las tareas pendientes para evitar crear una gran cantidad de mensajes pendientes con el mismo ID de grupo de mensajes. `MessageGroupId` es obligatorio para las colas FIFO. No puede utilizar esta opción para las colas estándar. Debe asociar un `MessageGroupId` que no esté vacío con un mensaje. Si no proporciona un `MessageGr oupId` , la acción genera un error.

Cuota	Descripción
	La longitud de `MessageGroupId` es de 128 caracteres. Valores válidos: caracteres alfanuméricos y signos de puntuación `(!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~)` .
Retención de mensajes	De forma predeterminada, un mensaje se conserva durante 4 días. El mínimo es 60 segundos (1 minuto). El máximo es 1 209 600 segundos (14 días).
Capacidad de procesamiento de mensajes	Las colas estándar admiten un número casi ilimitado de llamadas a la API por segundo, por acción de API (`SendMessage` , `ReceiveMessage` o `DeleteMessage`).
	Colas FIFO • Las colas FIFO admiten una cuota de 300 transacciones por segundo y por acción de la API (`SendMessage` , `ReceiveMessage` y `DeleteMessage`). • Si utiliza procesamiento por lotes, las colas FIFO admiten hasta 3000 mensajes por segundo y por método de la API (`SendMessage` , `ReceiveMessage` y `DeleteMessage`). Los 3000 mensajes por segundo representan 300 llamadas a la API, cada una con un lote de 10 mensajes.

Cuota	Descripción
	Alto rendimiento para las colas FIFO • Sin procesamiento por lotes (`SendMessage`, `ReceiveMessage` y `DeleteMessage`), el alto rendimiento de las colas FIFO procesa hasta 70 000 transacciones por segundo, por acción de la API en las regiones de Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón) y Europa (Irlanda). • Para las regiones de Este de EE. UU. (Ohio) y Europa (Fráncfort), el rendimiento predeterminado es de 18 000 transacciones por segundo por acción de la API. • Para las regiones Asia-Pacífico (Bombay), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney) y Asia-Pacífico (Tokio), el rendimiento predeterminado es de 9000 transacciones por segundo y por acción de la API. • En Europa (Londres) y América del Sur (São Paulo), el rendimiento predeterminado es de 4500 transacciones por segundo y por acción de la API. • Para obtener el máximo rendimiento, aumente el número de ID de grupo de mensajes que utiliza para los mensajes enviados sin procesamiento por lotes. • Puede aumentar el rendimiento hasta 700 000 mensajes por segundo mediante las API de procesamiento por lotes (`SendMessageBatch` y `DeleteMessageBatch`) en las regiones de Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón) y Europa (Irlanda). Los 700 000 mensajes por segundo representan 70 000 transacciones por segundo, cada una con un lote de 10 mensajes.

Cuota	Descripción
	Para las regiones de Europa (Fráncfort) y Este de EE. UU. (Ohio), puede alcanzar hasta 180 000 mensajes por segundo con las API de procesamiento por lotes. Los 180 000 mensajes por segundo representan 18 000 transacciones por segundo, cada una con un lote de 10 mensajes. Para las regiones de Asia-Pacífico (Bombay), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney) y Asia-Pacífico (Tokio), puede alcanzar hasta 90 000 mensajes por segundo con el procesamiento por lotes. Para lograr el máximo rendimiento al usar `SendMessageBatch` y `DeleteMessageBatch`, todos los mensajes de una solicitud por lotes deben usar el mismo ID de grupo de mensajes. • Para las regiones de Europa (Londres) y América del Sur (São Paulo), puede alcanzar hasta 45 000 mensajes por segundo con el procesamiento por lotes. Para lograr el máximo rendimiento al usar `SendMessageBatch` y `DeleteMessageBatch`, todos los mensajes de una solicitud por lotes deben usar el mismo ID de grupo de mensajes. • En todas AWS las demás regiones, el rendimiento máximo es de 2400 (sin procesamiento por lotes) o 24 000 (si se utiliza el procesamiento por lotes) mensajes por segundo, por acción de la API. • Para solicitar un aumento de la cuota por encima del límite regional, envía una solicitud de soporte. • Para obtener más información, consulte Particiones y distribución de datos para obtener alto rendimiento en las colas FIFO de SQS.
Temporizador de mensajes	El retraso predeterminado (mínimo) de un mensaje es de 0 segundos. El valor máximo es de 15 minutos.

Cuota	Descripción
Tamaño del mensaje	El tamaño mínimo de los mensajes es de 1 byte (1 carácter). El máximo es de 262 144 bytes (256 KiB). Para enviar mensajes de más de 256 KiB, puede utilizar la biblioteca de clientes extendida de Amazon SQS para Java y la biblioteca de clientes extendida de Amazon SQS para Python. Esta biblioteca le permite enviar un mensaje de Amazon SQS que contiene una referencia a una carga de mensajes de Amazon S3. El tamaño de carga máximo es 2 GB. **Note** Esta biblioteca ampliada solo funciona para clientes síncronos.
Tiempo de espera de visibilidad de los mensajes	El tiempo de espera de visibilidad predeterminado de un mensaje es de 30 segundos. El mínimo es de 0 segundos. El máximo es de 12 horas.
Información de políticas	La cuota máxima es 8192 bytes, 20 instrucciones, 50 entidades principales o 10 condiciones. Para obtener más información, consulte Cuotas de políticas de Amazon SQS.

Cuotas de políticas de Amazon SQS

En la siguiente tabla se muestran las cuotas relacionadas con las políticas.

Nombre	Máximo
Bytes	8 192
Condiciones	10

Nombre	Máximo
Entidades principales	50
Instrucciones	20
Acciones por instrucción	7

Características y capacidades de Amazon SQS

En Amazon SQS, se beneficia de las siguientes características y funciones básicas.

Temas

Uso de colas de letra muerta en Amazon SQS

Amazon SQS admite colas de cartas muertas (DLQ), a las que pueden dirigirse las colas de origen para los mensajes que no se procesan correctamente. Los DLQ son útiles para depurar las aplicaciones porque permiten aislar los mensajes no consumidos para determinar por qué el procesamiento no se ha realizado correctamente. Para obtener un rendimiento óptimo, se recomienda mantener la cola de origen y el DLQ dentro de la misma región. Cuenta de AWS Una vez que los mensajes estén en una cola de letra muerta, puedes:

- Examinar los registros para ver si hay excepciones que podrían haber causado el traslado de mensajes a una cola de mensajes fallidos.
- Analice el contenido de los mensajes transferidos a la cola de mensajes sin efecto para diagnosticar problemas con las aplicaciones.

- Determinar si ha concedido al consumidor tiempo suficiente para procesar los mensajes.
- Saque los mensajes de la cola de letra muerta mediante el redrive de la cola de letra muerta.

Primero debes crear una nueva cola antes de configurarla como cola de letra muerta. Para obtener información sobre la configuración de una cola de mensajes fallidos mediante la consola de Amazon SQS, consulte Aprenda a configurar una cola de cartas sin salida mediante la consola Amazon SQS. Para obtener ayuda con las colas de letra muerta, por ejemplo, cómo configurar una alarma para cualquier mensaje que se traslade a una cola de letra muerta, consulte. Crea alarmas para colas de cartas sin salida con Amazon CloudWatch

Uso de políticas para colas de cartas sin salida

Utilice una política de redrive para especificar el. `maxReceiveCount` `maxReceiveCount`Es el número de veces que un consumidor puede recibir un mensaje de una cola de origen antes de pasarlo a una cola de letra muerta. Por ejemplo, si `maxReceiveCount` se establece en un valor bajo, como 1, si no se recibe un mensaje, el mensaje pasará a la cola de mensajes sin respuesta. Para asegurarse de que su sistema es resiliente frente a los errores, establezca `maxReceiveCount` lo suficientemente alto como para permitir un número suficiente de reintentos.

La política de permiso de redireccionamiento especifica qué colas de origen pueden acceder a la cola de mensajes fallidos. Puede elegir entre permitir todas las colas de origen, permitir colas de origen específicas o denegar a todas las colas de origen el uso de la cola de letra muerta. La opción predeterminada permite que todas las colas de origen utilicen la cola de letras sin efecto. Si decide permitir colas específicas mediante la `byQueue` opción, puede especificar hasta 10 colas de origen utilizando la cola de origen Amazon Resource Name (ARN). Si especifica `denyAll`, la cola no se puede utilizar como una cola de mensajes fallidos.

Descripción de los períodos de retención de mensajes para las colas con letra muerta

En el caso de las colas estándar, la caducidad de un mensaje siempre se basa en su marca temporal original. Cuando un mensaje se mueve a una cola de mensajes fallidos, la marca temporal de la cola no se modifica. La `ApproximateAgeOfOldestMessage` métrica indica cuándo el mensaje pasó a la cola de mensajes sin salida, no cuándo se envió originalmente. Por ejemplo, supongamos que un mensaje pasa un día en la cola original antes de ser trasladado a una cola de mensajes fallidos. Si el periodo de retención de la cola de mensajes fallidos es de cuatro días, el mensaje se elimina de la cola de mensajes fallidos al cabo de tres días y `ApproximateAgeOfOldestMessage` es de

tres días. Por lo tanto, se recomienda establecer siempre un periodo de retención de una cola de mensajes fallidos superior al periodo de retención de la cola original.

Para las colas FIFO, la marca temporal de entrada se restablece cuando el mensaje se mueve a una cola de mensajes fallidos. La métrica `ApproximateAgeOfOldestMessage` indica cuándo el mensaje ha pasado a la cola de mensajes fallidos. En el mismo ejemplo anterior, el mensaje se elimina de la cola de mensajes fallidos al cabo de cuatro días y `ApproximateAgeOfOldestMessage` es de cuatro días.

Aprenda a configurar una cola de cartas sin salida mediante la consola Amazon SQS

Una cola de mensajes sin salida es una cola a la que pueden dirigirse las colas de origen para los mensajes que no se procesan correctamente. Para obtener más información, consulte Uso de colas de letra muerta en Amazon SQS .

Amazon SQS no crea la cola de mensajes fallidos automáticamente. Primero deberá crear la cola antes de utilizarla como una cola de mensajes fallidos. Para obtener instrucciones sobre cómo crear una cola para utilizarla como cola de cartas no escritas, consulte. Cree una cola con la consola Amazon SQS

La cola de mensajes fallidos de una cola FIFO también debe ser una cola FIFO. De igual manera, la cola de mensajes fallidos de una cola estándar también debe ser una cola estándar.

Al crear o editar una cola, puede configurar una cola de mensajes fallidos.

Configuración de una cola de mensajes fallidos para una cola existente (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. Seleccione una cola y elija Editar.
4. Desplácese hasta la sección Cola de mensajes fallidos y elija Activado.
5. Elija el nombre de recurso de Amazon (ARN) de una cola de mensajes fallidos existente que desee asociar a esta cola de origen.
6. Para configurar el número de veces que se puede recibir un mensaje antes de que se envíe a una cola de mensajes fallidos, defina en Recepciones máximas un valor comprendido entre 1 y 1000.
7. Cuando termine de configurar la cola de mensajes fallidos, elija Guardar.

Después de guardar la cola, la consola muestra la página Detalles de la cola. En la página Detalles, la pestaña Cola de mensajes fallidos muestra el ARN de Recepciones máximas y Cola de mensajes fallidos en la Cola de mensajes fallidos.

Aprenda a configurar un redrive de colas con letra muerta en Amazon SQS

Puedes utilizar la redirección de colas de cartas sin procesar para sacar los mensajes no consumidos de una cola de cartas sin salida existente. De forma predeterminada, el redireccionamiento de cola de mensajes fallidos mueve los mensajes de una cola de mensajes fallidos a una cola de origen. No obstante, también puede configurar cualquier otra cola estándar como destino de redireccionamiento si las colas son del mismo tipo. Por ejemplo, si la cola de mensajes fallidos es una cola FIFO, la cola de destino de redireccionamiento también debe ser una cola FIFO. Además, puede configurar la velocidad de redireccionamiento para establecer a qué velocidad Amazon SQS mueve los mensajes.

> **Note**
>
> Cuando un mensaje se mueve de una cola de FIFO a un DLQ de FIFO, el identificador de deduplicación del mensaje original se sustituirá por el identificador del mensaje original. Esto es para asegurarse de que la desduplicación de DLQ no impedirá el almacenamiento de dos mensajes independientes que casualmente comparten un ID de desduplicación.

Las colas con letra muerta redistribuyen los mensajes en el orden en que se reciben, empezando por el mensaje más antiguo. Sin embargo, la cola de destino ingiere los mensajes redirigidos, así como los mensajes nuevos de otros productores, según el orden en que los recibe. Por ejemplo, si un productor envía mensajes a una cola FIFO de origen y recibe simultáneamente mensajes redireccionados de una cola de correspondencia muerta, los mensajes redireccionados se entrelazarán con los nuevos mensajes del productor.

> **Note**
>
> La tarea de redireccionamiento restablece el periodo de retención. Todos los mensajes redirigidos se consideran mensajes nuevos con un mensaje nuevo y se asignan a mensajes redireccionados. `messageID` `enqueueTime`

Temas

- Configuración de un redrive de cola de letra muerta para una cola estándar existente mediante la API Amazon SQS
- Configuración de un redrive de cola de letra muerta para una cola estándar existente mediante la consola Amazon SQS
- Configuración de los permisos del redireccionamiento de cola de mensajes fallidos

Configuración de un redrive de cola de letra muerta para una cola estándar existente mediante la API Amazon SQS

Puedes configurar un redrive de colas con letra muerta mediante las acciones, y de la API: `SendMessageBatch` `ReceiveMessage` `DeleteMessageBatch`

Acción de la API	Descripción
`StartMessageMoveTask`	Inicia una tarea asincrónica para mover mensajes de una cola de origen especificada a una cola de destino especificada.
`ListMessageMoveTasks`	Obtiene las tareas de movimiento de mensajes más recientes (hasta diez) en una cola de origen específica.
`CancelMessageMoveTask`	Cancela una tarea de movimiento de mensajes especificada. Un movimiento de mensajes solo puede cancelarse cuando el estado actual es EN EJECUCIÓN.

Configuración de un redrive de cola de letra muerta para una cola estándar existente mediante la consola Amazon SQS

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. Elija el nombre de la cola que haya configurado como una cola de mensajes fallidos.
4. Seleccione Iniciar el redireccionamiento de la DLQ.

5. En Redireccionar configuración, para Destino del mensaje, realice una de las siguientes acciones:
   - Para redireccionar mensajes a su cola de origen, seleccione Redireccionar a las colas de origen.
   - Para redireccionar mensajes a otra cola, elija Redireccionar a un destino personalizado. A continuación, escriba el nombre de recurso de Amazon (ARN) de una cola de destino existente.
6. En Configuración de control de velocidad, elija una de las siguientes opciones:
   - Sistema optimizado: redireccionar la cola de mensajes fallidos al número máximo de mensajes por segundo.redireccionar la cola de mensajes fallidos al número máximo de mensajes por segundo.
   - Velocidad máxima personalizada: redireccionar mensajes de la cola de mensajes fallidos con una velocidad máxima personalizada de mensajes por segundo. La velocidad máxima permitida es de 500 mensajes por segundo.
     - Se recomienda empezar con un valor pequeño para la velocidad máxima personalizada y comprobar que la cola de origen no se satura de mensajes. A partir de ahí, aumente gradualmente el valor de la velocidad máxima personalizada, sin dejar de monitorear el estado de la cola de origen.
7. Cuando termine de configurar el redireccionamiento de la cola de mensajes fallidos, elija Redireccionar mensajes.

   > **Important**
   >
   > Amazon SQS no admite el filtrado y la modificación de mensajes mientras los redirecciona desde la cola de mensajes fallidos.
   > Una tarea de redireccionamiento de cola de mensajes fallidos puede ejecutarse un máximo de 36 horas. Amazon SQS admite un máximo de 100 tareas de redireccionamiento activas por cuenta.

8. Si desea cancelar la tarea de redireccionamiento de mensajes, en la página Detalles de la cola, elija Cancelar redireccionamiento de DLQ. Al cancelar un redireccionamiento de mensajes en curso, los mensajes que ya se hayan movido correctamente a su cola de destino de movimiento permanecerán en ella.

Configuración de los permisos del redireccionamiento de cola de mensajes fallidos

Puede conceder al usuario acceso a acciones específicas de la cola de mensajes fallidos si agrega permisos a la política. Los permisos mínimos necesarios para un redireccionamiento de cola de mensajes fallidos son los siguientes:

Permisos mínimos	Métodos de API necesarios
Inicio de un redirecci onamiento de mensajes	• Agregue `sqs:StartMessageMoveTask` , `sqs:ReceiveMessage` , `sqs:DeleteMessage` y `sqs:GetQueueAttributes` de la cola de mensajes fallidos. Si la cola de mensajes fallidos o la cola de origen están cifradas (conocida como cola SSE), también se necesita `kms:Decrypt` para cualquier clave de KMS que se haya utilizado para cifrar los mensajes. • Agregue `sqs:SendMessage` de la cola de destino. Si la cola de destino está cifrada, también se requieren `kms:GenerateDataKey` y `kms:Decrypt` .
Cancelación de un redirecci onamiento de mensajes en curso	• Agregue `sqs:CancelMessageMoveTask` , `sqs:ReceiveMessage` , `sqs:DeleteMessage` y `sqs:GetQueueAttributes` de la cola de mensajes fallidos. Si la cola de mensajes fallidos está cifrada (conocida como cola SSE), también se requiere `kms:Decrypt` .
Visualización del estado de movimiento de mensajes	• Agregue `sqs:ListMessageMoveTasks` y `sqs:GetQueueAttributes` de la cola de mensajes fallidos.

Configuración de los permisos de un par de colas cifradas (una cola de origen con una cola de mensajes fallidos)

Siga estos pasos para configurar los permisos mínimos de un redireccionamiento de cola de mensajes fallidos:

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Políticas.
3. Cree una política con los siguientes permisos y asóciela a su usuario o rol de IAM de inicio de sesión:

   - `sqs:StartMessageMoveTask`
   - `sqs:CancelMessageMoveTask`
   - `sqs:ListMessageMoveTasks`
   - `sqs:ListDeadLetterSourceQueues`
   - `sqs:ReceiveMessage`
   - `sqs:DeleteMessage`
   - `sqs:GetQueueAttributes`
   - El ARN de `Resource` de la cola de mensajes fallidos (por ejemplo, “arn:aws:sqs:*<DLQ_region>*:*<DLQ_accountId>*:*<DLQ_name>*”).
   - `sqs:SendMessage`
   - El `Resource` ARN de la cola de destino (por ejemplo, «arn:aws:sqs: *< DestQueue _region>: < _accountID>: < _name>* «). DestQueue DestQueue
   - `kms:Decrypt`: permite la acción de descifrado.
   - `kms:GenerateDataKey`
   - Los ARN de `Resource` de cualquier clave de cifrado de KMS que se haya utilizado para cifrar los mensajes de la cola de origen (por ejemplo, “arn:aws:kms:*<region>*:*<accountId>*:key/*<keyId_used to encrypt the message body>*”).
   - El ARN de recurso de la clave de cifrado de KMS que se utiliza para la cola de destino de redireccionamiento (por ejemplo, “arn:aws:kms:*<region>*:*<accountId>*:key/ <*keyId_utilizado para la cola de destino*>").

   Su política de acceso debe ser similar a la siguiente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "sqs:StartMessageMoveTask",
        "sqs:CancelMessageMoveTask",
        "sqs:ListMessageMoveTasks",
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes",
        "sqs:ListDeadLetterSourceQueues"
      ],
      "Resource": "arn:aws:sqs:<DLQ_region>:<DLQ_accountId>:<DLQ_name>"
    },
    {
      "Effect": "Allow",
      "Action": "sqs:SendMessage",
      "Resource":
 "arn:aws:sqs:<DestQueue_region>:<DestQueue_accountId>:<DestQueue_name>"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:<region>:<accountId>:key/<keyId>"
    }
  ]
}
```

Configuración de los permisos mediante un par de colas no cifradas (una cola de origen con una cola de mensajes fallidos)

Siga estos pasos para configurar los permisos mínimos de una cola de mensajes fallidos no cifrada. Los permisos mínimos requeridos son recibir, eliminar y obtener atributos de la cola de mensajes fallidos, y enviar atributos a la cola de origen.

1. Inicie AWS Management Console sesión en la consola de IAM y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Políticas.
3. Cree una política con los siguientes permisos y asóciela a su usuario o rol de IAM de inicio de sesión:

- `sqs:StartMessageMoveTask`
- `sqs:CancelMessageMoveTask`
- `sqs:ListMessageMoveTasks`
- `sqs:ListDeadLetterSourceQueues`
- `sqs:ReceiveMessage`
- `sqs:DeleteMessage`
- `sqs:GetQueueAttributes`
- El ARN de `Resource` de la cola de mensajes fallidos (por ejemplo, “arn:aws:sqs:*<DLQ_region>*:*<DLQ_accountId>*:*<DLQ_name>*”).
- `sqs:SendMessage`
- *El Resource ARN de la cola de destino (por ejemplo, «arn:aws:sqs: < DestQueue _region>: < _accountID>: < _name> «). DestQueue DestQueue*

Su política de acceso debe ser similar a la siguiente:

```
{
"Version": "2012-10-17",
  "Statement": [
    {
"Effect": "Allow",
      "Action": [
        "sqs:StartMessageMoveTask",
        "sqs:CancelMessageMoveTask",
        "sqs:ListMessageMoveTasks",
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes",
        "sqs:ListDeadLetterSourceQueues"
      ],
      "Resource": "arn:aws:sqs:<DLQ_region>:<DLQ_accountId>:<DLQ_name>"
    },
    {
"Effect": "Allow",
      "Action": "sqs:SendMessage",
      "Resource":
 "arn:aws:sqs:<DestQueue_region>:<DestQueue_accountId>:<DestQueue_name>"
    }
```

```
  ]
}
```

CloudTrail requisitos de actualización y permiso para la reactivación de colas de cartas muertas de Amazon SQS

El 8 de junio de 2023, Amazon SQS introdujo el redrive de cola de letras muertas (DLQ) para AWS el SDK y (CLI). AWS Command Line Interface Esta capacidad se suma al redrive DLQ ya compatible con la consola. AWS Si ya utilizaste la AWS consola para reimprimir los mensajes en cola con letra muerta, es posible que te afecten los siguientes cambios:

- CloudTrail cambio de nombre de un evento para reconducir las colas con letra muerta
- Permisos actualizados para el redireccionamiento de la cola de mensajes fallidos

CloudTrail cambio de nombre de eventos

El 15 de octubre de 2023, cambiarán los nombres de los CloudTrail eventos de redrive de colas con letra muerta en la consola Amazon SQS. Si ha configurado alarmas para estos CloudTrail eventos, debe actualizarlas ahora. Los nuevos nombres de CloudTrail eventos de DLQ redrive son los siguientes:

Nombre de evento anterior	Nombre de evento nuevo
`CreateMoveTask`	`StartMessageMoveTask`
`CancelMoveTask`	`CancelMessageMoveTask`

Permisos actualizados

Incluido con el lanzamiento del SDK y CLI, Amazon SQS también ha actualizado los permisos de cola para el redireccionamiento de DLQ a fin de cumplir con las prácticas recomendadas de seguridad. Utilice los siguientes tipos de permisos de cola para redireccionar mensajes de las DLQ.

1. Permisos basados en acciones (actualización de las acciones de la API de DLQ)
2. Permisos de política de Amazon SQS
3. Política de permisos que utiliza caracteres comodín sqs:*

> **Important**
>
> Para utilizar el redireccionamiento de DLQ para SDK o CLI, es necesario disponer de una política de permisos de redireccionamiento de DLQ que coincida con una de las opciones anteriores.

Si sus permisos de cola para el redireccionamiento de DLQ no coinciden con alguna de las opciones anteriores, deberá actualizarlos antes del 31 de agosto de 2023. Desde ahora hasta el 31 de agosto de 2023, su cuenta podrá redireccionar mensajes con los permisos que haya configurado a través de la consola de AWS solo en las regiones en las que haya utilizado anteriormente el redireccionamiento de DLQ. Por ejemplo, supongamos que tiene “Cuenta A” tanto en us-east-1 como en eu-west-1. La «Cuenta A» se utilizó para redirigir los mensajes de la AWS consola en us-east-1 antes del 8 de junio de 2023, pero no en eu-west-1. Entre el 8 de junio de 2023 y el 31 de agosto de 2023, si los permisos de la política de la «Cuenta A» no coinciden con una de las opciones anteriores, solo se pueden usar para redirigir los mensajes de la AWS consola en us-east-1 y no en eu-west-1.

> **Important**
>
> Si sus permisos de redireccionamiento de DLQ no coinciden con una de estas opciones después del 31 de agosto de 2023, su cuenta ya no podrá redireccionar mensajes de DLQ mediante la consola de AWS .
> Sin embargo, si utilizaste la función de reaccionamiento de DLQ en la AWS consola durante agosto de 2023, dispones de una prórroga hasta el 15 de octubre de 2023 para adoptar los nuevos permisos según una de estas opciones.
> Para obtener más información, consulte the section called “Identificación de las políticas afectadas”.

A continuación, se muestran ejemplos de permisos de cola para cada opción de redireccionamiento de DLQ. Cuando se utilizan colas cifradas del lado del servidor (SSE), se requiere el permiso de clave correspondiente AWS KMS .

Basado en acciones

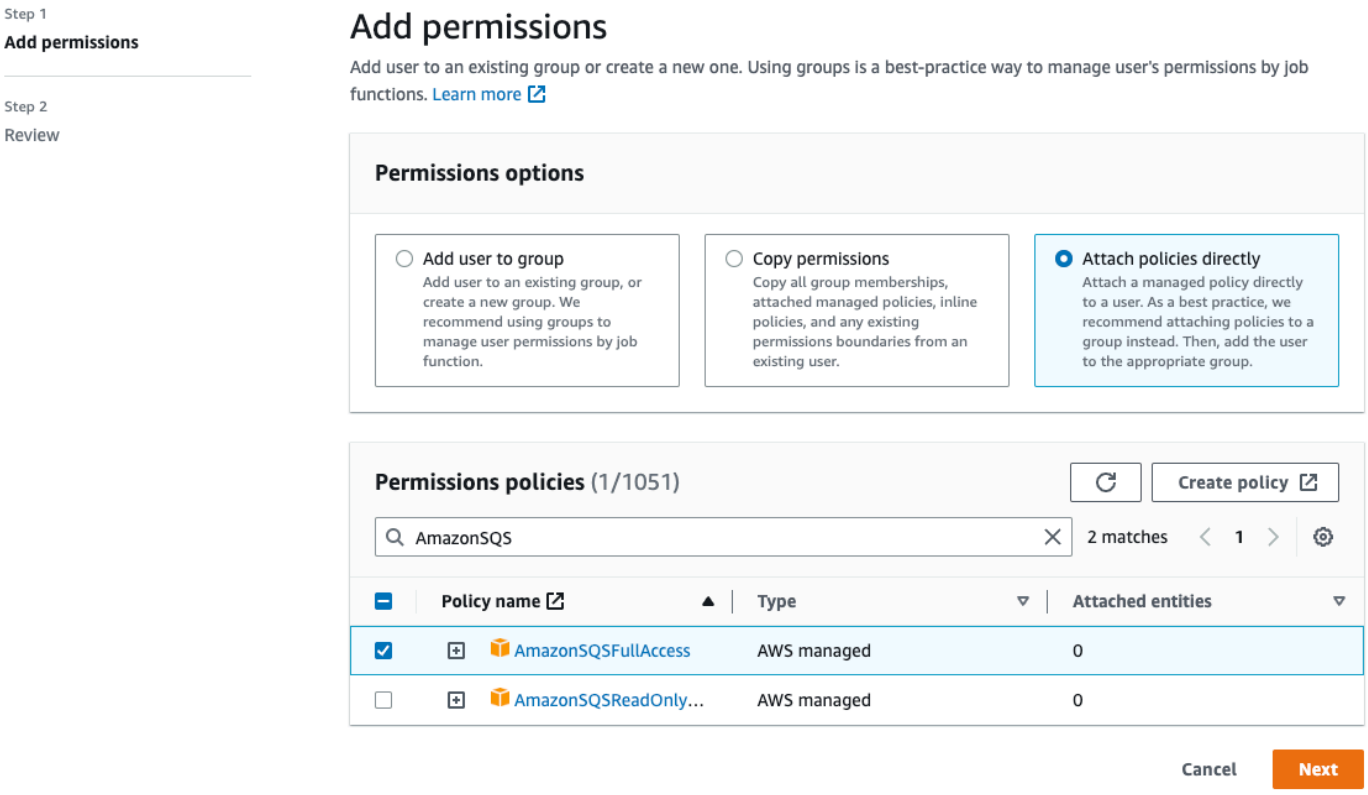
```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes",
        "sqs:StartMessageMoveTask",
        "sqs:ListMessageMoveTasks",
        "sqs:CancelMessageMoveTask"
      ],
      "Resource": "arn:aws:sqs:<DLQ_region>:<DLQ_accountId>:<DLQ_name>"
    },
    {
      "Effect": "Allow",
      "Action": "sqs:SendMessage",
      "Resource":
 "arn:aws:sqs:<DestQueue_region>:<DestQueue_accountId>:<DestQueue_name>"
    }
  ]
}
```

Política administrada

Las siguientes políticas administradas contienen los permisos actualizados necesarios:

- AmazonSQS FullAccess: incluye las siguientes tareas de redireccionamiento de colas con letra muerta: iniciar, cancelar y enumerar.
- AmazonSQS ReadOnly Access: proporciona acceso de solo lectura e incluye la tarea de regenerar listas de colas con letra muerta.



Política de permisos que utiliza caracteres comodín sqs*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sqs:*",
      "Resource": "*"
    }
  ]
}
```

Identificación de las políticas afectadas

Si utiliza políticas gestionadas por el cliente (CMP), puede utilizar AWS CloudTrail un IAM para identificar las políticas afectadas por la actualización de los permisos de colas.

> **Note**
>
> Si utiliza `AmazonSQSFullAccess` y `AmazonSQSReadOnlyAccess`, no es necesario realizar ninguna otra acción.

1. Inicie sesión en la consola. AWS CloudTrail
2. En la página Historial de eventos, en Buscar atributos, utilice el menú desplegable para seleccionar Nombre del evento. A continuación, busque `CreateMoveTask`.
3. Elija un evento para abrir la página Detalles. En la sección Registros de eventos, recupere `UserName` o `RoleName` del ARN de `userIdentity`.
4. Inicie sesión en la consola de IAM.
   - Para usuarios, seleccione Usuarios. Seleccione el usuario con el `UserName` identificado en el paso anterior.
   - Para roles, seleccione Roles. Busque el usuario con el `RoleName` identificado en el paso anterior.
5. En la página Detalles, en la sección Permisos, revise las políticas que tengan el prefijo `sqs:` en `Action` o revise las políticas que tengan la cola de Amazon SQS definida en `Resource`.

Crea alarmas para colas de cartas sin salida con Amazon CloudWatch

Puedes configurar una alarma para cualquier mensaje que se traslade a una cola de letra muerta con Amazon CloudWatch y la métrica. `ApproximateNumberOfMessagesVisible` Para obtener más información, consulte Creación de CloudWatch alarmas para las métricas de Amazon SQS. Cuando recibas una alerta de que los mensajes se han enviado a la cola de mensajes sin salida, puedes revisarlos mediante sondeos para recibirlos.

Metadatos de mensajes para Amazon SQS

Puede utilizar atributos de mensajes para adjuntar metadatos personalizados a los mensajes de Amazon SQS para sus aplicaciones. Puede utilizar atributos del sistema de mensajes para almacenar metadatos para otros servicios de AWS , como AWS X-Ray.

Temas

- Atributos de mensajes de Amazon SQS

- Atributos del sistema de mensajes de Amazon SQS

Atributos de mensajes de Amazon SQS

Amazon SQS le permite incluir metadatos estructurados (como marcas temporales, datos geoespaciales, firmas e identificadores) con los mensajes mediante atributos de mensaje. Cada mensaje puede tener hasta 10 atributos. Los atributos de los mensajes son opcionales y están separados del cuerpo del mensaje (sin embargo, se envían con él). El consumidor puede utilizar los atributos de mensajes para controlar un mensaje de una forma concreta sin tener que procesar el cuerpo del mensaje en primer lugar. Para obtener información sobre cómo enviar mensajes con atributos mediante la consola de Amazon SQS, consulte Envío de un mensaje con atributos.

> **Note**
>
> No confunda los atributos de los mensajes con los atributos del sistema de mensajes: si bien puede usar los atributos de los mensajes para adjuntar metadatos personalizados a los mensajes de Amazon SQS para sus aplicaciones, puede usar los atributos del sistema de mensajes para almacenar metadatos para otros AWS servicios, como. AWS X-Ray

Temas

- Componentes de atributos de mensajes
- Tipos de datos de atributos de mensajes
- Cálculo del resumen del mensaje MD5 para atributos de mensajes

Componentes de atributos de mensajes

> **Important**
>
> Todos los componentes de un atributo de mensaje están incluidos en la restricción de tamaño de mensaje de 256 KB.
> `Name`, `Type`, `Value` y el cuerpo del mensaje no deben estar vacíos ni ser null.

Cada atributo de mensaje consta de los siguientes componentes:

- Nombre: el nombre del atributo de mensaje puede contener los siguientes caracteres: A-Z, a-z, 0-9, guion bajo (_), guion (-) y punto (.). Se aplican las siguientes restricciones:
  - Puede tener hasta 256 caracteres
  - No puede comenzar por `AWS.` o `Amazon.` (ni ninguna variación de mayúsculas y minúsculas)
  - Distingue entre mayúsculas y minúsculas
  - Debe ser único entre todos los nombres de atributo del mensaje
  - No debe comenzar ni terminar por un punto
  - No debe tener puntos seguidos
- Tipo: el tipo de datos de atributo de mensaje. Los tipos admitidos son `String`, `Number` y `Binary`. También puede agregar información personalizada para cualquier tipo de datos. El tipo de datos tiene las mismas restricciones que el cuerpo del mensaje (para obtener más información, consulte `SendMessage` en la Referencia de la API de Amazon Simple Queue Service). Además, se aplican las siguientes restricciones:
  - Puede tener hasta 256 caracteres
  - Distingue entre mayúsculas y minúsculas
- Valor: el valor del atributo de mensaje. Para los tipos de datos `String`, los valores de atributo tienen las mismas restricciones el cuerpo del mensaje.

Tipos de datos de atributos de mensajes

Los tipos de datos de atributo de mensaje indican a Amazon SQS cómo tratar los correspondientes valores de atributo de mensaje. Por ejemplo, si el tipo es `Number`, Amazon SQS valida valores numéricos.

Amazon SQS admite los tipos de datos lógicos `String`, `Number` y `Binary` con etiquetas de tipo de datos personalizadas opcionales con el formato *`.custom-data-type`*.

- Cadena: los atributos `String` pueden almacenar texto Unicode mediante cualquier carácter XML válido.
- Número: los atributos `Number` pueden almacenar valores numéricos positivos o negativos. Un número puede tener hasta 38 dígitos de precisión y puede estar comprendido entre 10^-128 y 10^+126.

> **Note**
>
> Amazon SQS quita los ceros al principio y al final.

- Binario: los atributos binarios permiten almacenar datos binarios de cualquier índole, como datos comprimidos, datos cifrados o imágenes.
- Personalizado: para crear un tipo de datos personalizado, añada la etiqueta custom-type a cualquier tipo de datos. Por ejemplo:
  - `Number.byte`, `Number.short`, `Number.int` y `Number.float` pueden ayudarle a distinguir entre tipos de números.
  - `Binary.gif` y `Binary.png` pueden ayudarle a distinguir entre tipos de archivos.

> **Note**
>
> Amazon SQS no interpreta, valida ni utiliza los datos añadidos.
> La etiqueta custom-type tiene las mismas restricciones que el cuerpo del mensaje.

Cálculo del resumen del mensaje MD5 para atributos de mensajes

Si usa el AWS SDK for Java, puede omitir esta sección. La clase `MessageMD5ChecksumHandler` del SDK para Java admite los resúmenes de mensajes MD5 para los atributos de mensajes de Amazon SQS.

Si utiliza la API Query o uno de los AWS SDK que no admite resúmenes de mensajes de MD5 para los atributos de mensajes de Amazon SQS, debe seguir las siguientes pautas para realizar el cálculo de los resúmenes de mensajes de MD5.

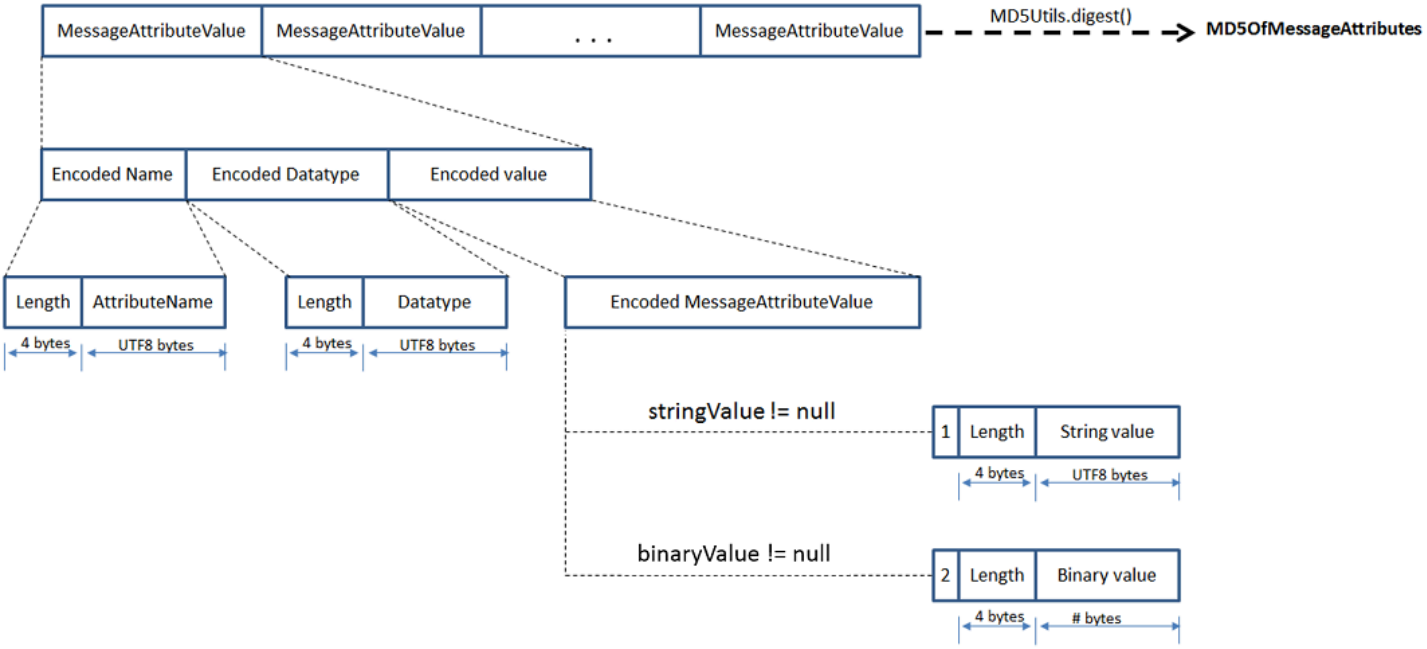
> **Note**
>
> Incluya siempre los sufijos de los tipos de datos personalizados en el cálculo del resumen de mensajes MD5.

Información general

A continuación se ofrece información general del algoritmo del cálculo del resumen del mensaje MD5:

1. Ordenar todos los atributos de mensajes por nombre en orden ascendente.
2. Codificar las partes individuales de cada atributo (`Name`, `Type` y `Value`) en un búfer.
3. Calcular el resumen de mensaje de todo el búfer.

En el siguiente diagrama se muestra la codificación del resumen del mensaje MD5 para un único atributo de mensaje:



Codificación de un único atributo de mensaje de Amazon SQS

1. Codifique el nombre: la longitud (4 bytes) y los bytes UTF-8 del nombre.
2. Codifique el tipo de datos: la longitud (4 bytes) y los bytes UTF-8 del tipo de datos.
3. Codifique el tipo de transporte (`String` o `Binary`) del valor (1 byte).

> **Note**
>
> Los tipos de datos lógicos `String` y `Number` utilizan el tipo de transporte `String`. El tipo de datos lógicos `Binary` utiliza el tipo de transporte `Binary`.

a. Para el tipo de transporte `String`, codifique 1.

b. Para el tipo de transporte `Binary`, codifique 2.

4. Codifique el valor del atributo.

a. Para el tipo de transporte `String`, codifique el valor del atributo: la longitud (4 bytes) y los bytes UTF-8 del valor.

b. En el tipo de transporte `Binary`, codifique el valor del atributo: la longitud (4 bytes) y los bytes sin formato del valor.

Atributos del sistema de mensajes de Amazon SQS

Aunque puede utilizar atributos de mensajes para adjuntar metadatos personalizados a mensajes de Amazon SQS para sus aplicaciones, puede utilizar atributos del sistema de mensajes para almacenar metadatos para otros servicios de AWS , como AWS X-Ray. Para obtener más información, consulte el parámetro de solicitud `MessageSystemAttribute` de las acciones de la API `SendMessage` y `SendMessageBatch`, el atributo `AWSTraceHeader` de la acción de la API `ReceiveMessage` y el tipo de datos `MessageSystemAttributeValue` en la Referencia de la API de Amazon Simple Queue Service.

Los atributos del sistema de mensajes se estructuran exactamente igual que los atributos del mensaje, con las siguientes excepciones:

- En la actualidad, el único atributo del sistema de mensajes admitido es `AWSTraceHeader`. Su tipo `String` y su valor deben ser una cadena de encabezado de AWS X-Ray rastreo con el formato correcto.
- El tamaño de un atributo del sistema de mensajes no cuenta para el tamaño total de un mensaje.

Recursos necesarios para procesar mensajes de Amazon SQS

Para ayudarlo a calcular los recursos que necesita para procesar los mensajes que hay en cola, Amazon SQS puede determinar el número aproximado de mensajes retrasados, visibles y no visibles de una cola. Para obtener más información acerca de la visibilidad, consulte Tiempo de espera de visibilidad de Amazon SQS.

> **Note**
>
> En el caso de las colas estándar, el resultado es aproximado debido a la arquitectura distribuida de Amazon SQS. En la mayoría de los casos, el recuento debe aproximarse al número real de mensajes de la cola.
> Para colas FIFO, el resultado es exacto.

En la tabla siguiente se muestra el nombre de atributo que se utilizará con la acción `GetQueueAttributes`:

Tarea	Nombre de atributo
Obtenga el número aproximado de mensajes disponibles para recuperar de la cola.	`ApproximateNumberOfMessages Visible`
Obtenga el número aproximado de mensajes de la cola que se retrasan y no están disponibl es para su lectura inmediata. Esto puede ocurrir cuando la cola está configurada como una cola de retraso o cuando se ha enviado un mensaje con un parámetro de retraso.	`ApproximateNumberOfMessages Delayed`
Obtenga el número aproximado de mensajes que se encuentran en tránsito. Se considera que los mensajes están en tránsito si se han enviado a un cliente pero aún no se han eliminado o aún no han llegado al final de su periodo de visibilidad.	`ApproximateNumberOfMessages NotVisible`

Paginación de colas de listas

Los métodos de API `listQueues` y `listDeadLetterQueues` admiten controles de paginación opcionales. De forma predeterminada, estos métodos de API devuelven hasta 1000 colas en el mensaje de respuesta. Puede configurar el parámetro `MaxResults` para que devuelva menos resultados en cada respuesta.

Establezca el parámetro `MaxResults` en la solicitud `listQueues` o `listDeadLetterQueues` para especificar el número máximo de resultados que se deben devolver en la respuesta. Si no establece `MaxResults`, la respuesta incluye un máximo de 1000 resultados y el valor de `NextToken` en la respuesta es nulo.

Si establece `MaxResults`, la respuesta incluye un valor para `NextToken` si hay resultados adicionales que mostrar. Use `NextToken` como parámetro en su próxima solicitud a `listQueues` para recibir la siguiente página de resultados. Si no hay resultados adicionales para mostrar, el valor de `NextToken` de la respuesta es nulo.

Etiquetas de asignación de costos de Amazon SQS

Para organizar e identificar las colas de Amazon SQS para la asignación de costos, puede agregar etiquetas de metadatos que identifiquen el propósito, el propietario o el entorno de una cola. Esto es útil especialmente cuando dispone de muchas colas. Para configurar las etiquetas con la consola de Amazon SQS, consulte the section called “Configuración de etiquetas para una cola”

Puede usar etiquetas de asignación de costos para organizar su AWS factura y reflejar su propia estructura de costos. Para ello, inscríbase para que su Cuenta de AWS factura incluya claves de etiquetas y valores. Para obtener más información, consulte Configuración de un informe de asignación de costos mensual en la Guía del usuario de AWS Billing .

Cada etiqueta consta de un par clave-valor, que usted define. Por ejemplo, puede identificar fácilmente sus colas de producción y prueba si etiqueta sus colas de la siguiente forma:

Queue	Clave	Valor
`MyQueueA`	`QueueType`	`Production`
`MyQueueB`	`QueueType`	`Testing`

> **Note**
>
> Cuando utilice etiquetas de cola, tenga en cuenta las siguientes directrices:
>
> - No recomendamos agregar más de 50 etiquetas a una cola. El etiquetado admite caracteres Unicode en UTF-8.

- Las etiquetas no tienen ningún significado semántico. Amazon SQS interpreta las etiquetas como cadenas de caracteres.
- Las etiquetas distinguen entre mayúsculas y minúsculas.
- Una etiqueta nueva con una clave idéntica a la de una etiqueta existente sobrescribe la existente.
- Las acciones de etiquetado están limitadas a 30 TPS cada una. Cuenta de AWS Si su aplicación requiere un mayor rendimiento, envíe una solicitud.

Para obtener una lista completa de restricciones de etiqueta, consulte Cuotas.

Sondeos cortos y largos de Amazon SQS

Amazon SQS ofrece opciones de sondeos cortos y largos para recibir mensajes de una cola. Tenga en cuenta los requisitos de capacidad de respuesta y rentabilidad de su aplicación al elegir entre estas dos opciones de sondeo:

- Sondeo breve (predeterminado): la `ReceiveMessage`solicitud consulta un subconjunto de servidores (según una distribución aleatoria ponderada) para encontrar los mensajes disponibles y envía una respuesta inmediata, incluso si no se encuentra ningún mensaje.
- Sondeo prolongado: `ReceiveMessage`consulta los mensajes en todos los servidores y envía una respuesta una vez que haya al menos un mensaje disponible, hasta el máximo especificado. Solo se envía una respuesta vacía si finaliza el tiempo de espera del sondeo. Esta opción puede reducir la cantidad de respuestas vacías y, potencialmente, reducir los costos.

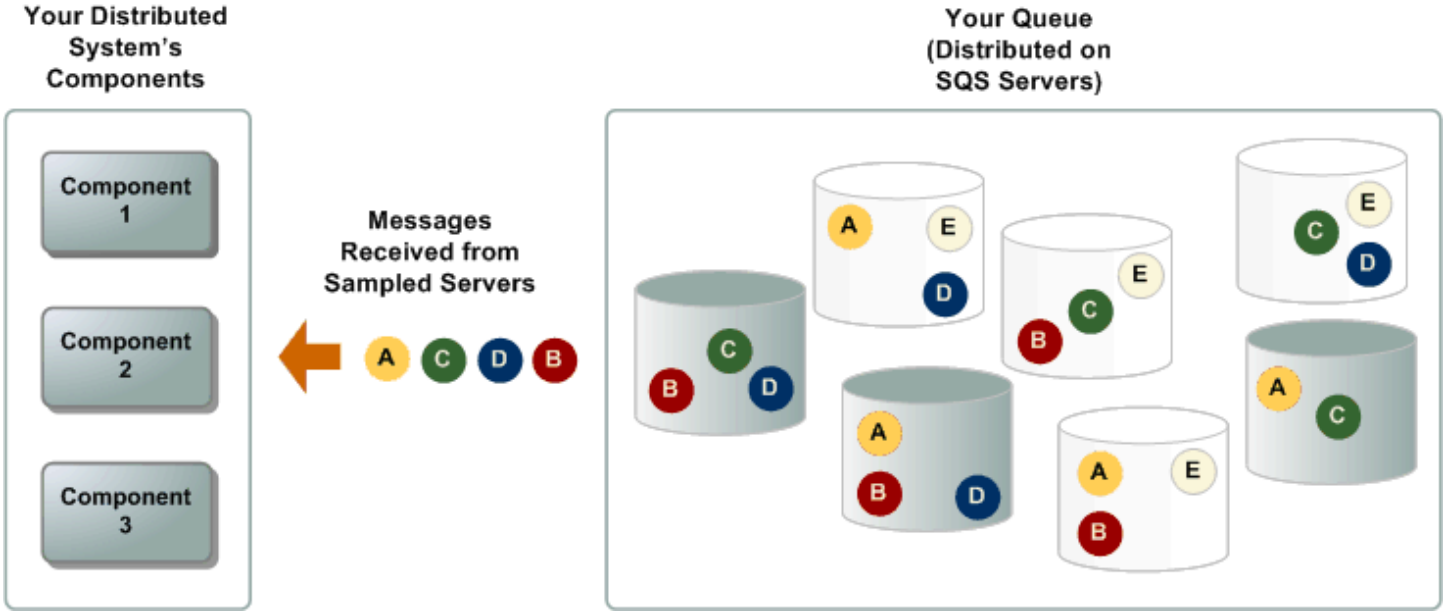
En las siguientes secciones se explican los detalles de los sondeos cortos y largos.

Temas

Consumo de mensajes mediante sondeo corto

Cuando consume mensajes de una cola (FIFO o estándar) mediante sondeos cortos, Amazon SQS toma muestras de un subconjunto de sus servidores (en función de una distribución aleatoria ponderada) y devuelve los mensajes únicamente de esos servidores. Por tanto, una solicitud `ReceiveMessage` determinada podría no devolver todos los mensajes. Sin embargo, si tiene menos de 1 000 mensajes en la cola, una solicitud posterior devolverá sus mensajes. Si sigue consumiendo mensajes de sus colas, Amazon SQS muestrea todos sus servidores y se reciben todos los mensajes.

En el siguiente diagrama se muestra el comportamiento de sondeo corto de los mensajes devueltos de una cola estándar después de que uno de los componentes del sistema realice una solicitud de recepción. Amazon SQS muestrea varios de sus servidores (en gris) y devuelve los mensajes A, C, D y B de estos servidores. El mensaje E no se devuelve en esta solicitud concreta, pero se devuelve en una solicitud posterior.



Consumo de mensajes mediante sondeo largo

Cuando el tiempo de espera de la acción de la API `ReceiveMessage` es superior a 0, se está realizando un sondeo largo. El tiempo máximo de espera de sondeo es de 20 segundos. El sondeo largo ayuda a reducir el costo de uso de Amazon SQS al eliminar el número de respuestas vacías (cuando no hay ningún mensaje disponible para una solicitud `ReceiveMessage`) y las falsas respuestas vacías (cuando los mensajes están disponibles en la cola, pero no se incluyen en una respuesta). Para obtener información sobre cómo habilitar el sondeo largo para una cola nueva o

existente mediante la consola de Amazon SQS, consulte Configuración de los parámetros de cola mediante la consola Amazon SQS. Para obtener las prácticas recomendadas, consulte Configuración del sondeo largo.

El sondeo largo ofrece los siguientes beneficios:

- Reducción de las respuestas vacías al permitir que Amazon SQS espere hasta que haya un mensaje disponible en una cola antes de enviar una respuesta. A menos que la conexión agote el tiempo de espera, la respuesta a la solicitud `ReceiveMessage` contiene al menos uno de los mensajes disponibles, hasta el número máximo de mensajes especificado en la acción `ReceiveMessage`. En contados casos, es posible que reciba respuestas vacías incluso cuando una cola aún contenga mensajes, sobre todo si especifica un valor bajo para el parámetro `ReceiveMessageWaitTimeSeconds`.
- Reduzca las falsas respuestas vacías consultando todos (en lugar de un subconjunto) los servidores de Amazon SQS.
- Devolución de mensajes en cuanto estén disponibles.

Para obtener información sobre cómo confirmar que una cola está vacía, consulte Confirmación de que una cola de Amazon SQS está vacía.

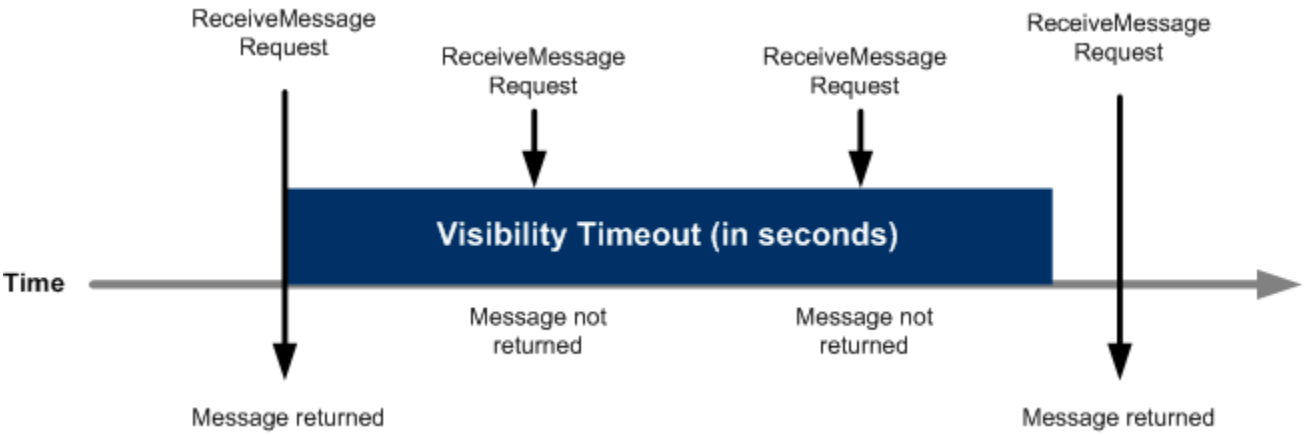
Diferencias entre el sondeo corto y el sondeo largo

El sondeo corto se produce cuando el parámetro `WaitTimeSeconds` de una solicitud `ReceiveMessage` está establecido en `0` de una de las dos formas siguientes:

- La llamada a `ReceiveMessage` establece `WaitTimeSeconds` en `0`.
- La llamada a `ReceiveMessage` no establece `WaitTimeSeconds` y el atributo de cola `ReceiveMessageWaitTimeSeconds` está establecido en `0`.

Tiempo de espera de visibilidad de Amazon SQS

Cuando un consumidor recibe y procesa un mensaje de una cola, el mensaje permanece en la cola. Amazon SQS no elimina automáticamente el mensaje. Dado que Amazon SQS es un sistema distribuido, no garantiza que el consumidor reciba realmente el mensaje (por ejemplo, debido a un problema de conectividad o a un problema en la aplicación del consumidor). Por tanto, el consumidor debe eliminar el mensaje de la cola después de recibirlo y procesarlo.



Inmediatamente después de recibirse un mensaje, este permanece en la cola. Para evitar que otros consumidores vuelvan a procesar el mensaje, Amazon SQS establece un tiempo de espera de visibilidad, que es un periodo de tiempo durante el cual Amazon SQS impide que otros consumidores reciban y procesen el mensaje. El tiempo de espera de visibilidad predeterminado de un mensaje es de 30 segundos. El mínimo es de 0 segundos. El máximo es de 12 horas. Para obtener información sobre la configuración del tiempo de espera de visibilidad de una cola mediante la consola, consulte Configuración de los parámetros de cola mediante la consola Amazon SQS.

> **Note**
>
> En las colas estándar, el tiempo de espera de visibilidad no es una garantía de que un mensaje no se recibirá dos veces. Para obtener más información, consulte Una t-least-once entrega.
>
> Las colas FIFO permiten al productor o al consumidor efectuar múltiples reintentos:
>
> - Si el productor detecta una acción `SendMessage` con error, puede reintentar el envío tantas veces como sea necesario, mediante el mismo ID de desduplicación de mensajes. Si se supone que el productor recibe al menos un acuse de recibo antes de que caduque el intervalo de desduplicación, los múltiples reintentos no afectan al orden de los mensajes ni generan duplicados.
> - Si el consumidor detecta una acción `ReceiveMessage` con error, puede volver a intentarlo tantas veces como sea necesario, mediante el mismo ID de intento de solicitud de recepción. Si se supone que el consumidor recibe al menos un acuse de recibo antes de que caduque el tiempo de visibilidad, los múltiples reintentos no afectan al orden de los mensajes.

- Cuando reciba un mensaje con un ID de grupo de mensajes, no se devolverán más mensajes para el mismo ID de grupo de mensajes a menos que elimine el mensaje o este se haga visible.

Temas

Mensajes en tránsito

Un mensaje de Amazon SQS tiene tres estados básicos:

1. Un productor lo ha enviado a una lista de espera.
2. Un consumidor lo ha recibido de la cola.
3. Se ha eliminado de la cola.

Se considera que un mensaje está almacenado después de que un productor lo envíe a una cola, pero un consumidor aún no lo haya recibido de la cola (es decir, entre los estados 1 y 2). No hay una cuota en cuanto al número de mensajes almacenados. Se considera que un mensaje está en tránsito después de que un consumidor lo haya recibido de una cola, pero aún no se haya eliminado de ella (es decir, entre los estados 2 y 3). Hay una cuota en cuanto al número de mensajes en tránsito.

Important

Las cuotas que se aplican a los mensajes en tránsito no guardan relación con el número ilimitado de mensajes almacenados.

En la mayoría de las colas estándar (según el tráfico de colas y los mensajes atrasados), puede haber un máximo de aproximadamente 120 000 mensajes en tránsito (recibidos de una cola por un consumidor, pero que aún no se han eliminado de la cola). Si alcanza esta cuota mientras utiliza sondeos cortos, Amazon SQS devuelve el mensaje de error `OverLimit`. Si utiliza sondeos largos,

Amazon SQS no devuelve ningún mensaje de error. Para evitar llegar a la cuota, conviene eliminar los mensajes de la cola una vez procesados. También puede aumentar el número de las colas que usa para procesar los mensajes. Para solicitar un aumento de la cuota, envíe una solicitud de soporte técnico.

En el caso de las colas FIFO, puede haber un máximo de 20 000 mensajes en tránsito (recibidos de una cola por un consumidor, pero aún no eliminados de la cola). Si alcanza esta cuota, Amazon SQS no devuelve ningún mensaje de error.

> **Important**
>
> Cuando se trabaja con colas FIFO, se producirá un error en las operaciones de `DeleteMessage` si la solicitud se recibe fuera del intervalo de tiempo de visibilidad. Si el tiempo de espera de visibilidad es de 0 segundos, el mensaje debe eliminarse en el mismo milisegundo en que se envió o se considerará abandonado. Esto puede provocar que Amazon SQS incluya mensajes duplicados en la misma respuesta a una operación de `ReceiveMessage` si el parámetro `MaxNumberOfMessages` es mayor que 1. Para obtener más información, consulte Cómo funciona la API FIFO de Amazon SQS.

Configuración del tiempo de espera de visibilidad

El tiempo de espera de visibilidad comienza cuando Amazon SQS devuelve un mensaje. Durante este tiempo, el consumidor procesa y elimina el mensaje. Sin embargo, si se produce un error en el consumidor, se elimina el mensaje y su sistema no llama a la `DeleteMessage` acción para dicho mensaje antes de que expire el tiempo de espera de visibilidad, el mensaje se hace visible para otros consumidores y se recibe nuevamente. Si un mensaje solo se debe recibir una vez, el consumidor debe eliminarlo dentro de la duración del tiempo de espera de visibilidad.

Cada cola de Amazon SQS posee un tiempo de espera de visibilidad predeterminado de 30 segundos. Puede cambiar esta configuración para toda la cola. Normalmente, debe definir el tiempo de espera de visibilidad como el tiempo máximo que tarda su aplicación en procesar y eliminar un mensaje de la cola. Cuando reciba mensajes, también puede establecer un tiempo de espera de visibilidad especial para los mensajes devueltos sin cambiar el tiempo de espera general de la cola. Para obtener más información, consulte las prácticas recomendadas en la sección Procesamiento de los mensajes a tiempo.

Si no sabe cuánto tiempo tarda en procesarse un mensaje, cree un latido para su proceso consumidor: especifique el tiempo de espera de visibilidad inicial (por ejemplo, dos minutos) y, a

continuación, mientras su consumidor siga trabajando en el mensaje, continúe ampliando el tiempo de espera de visibilidad en dos minutos cada minuto.

> **Important**
>
> El tiempo máximo de visibilidad es de 12 horas desde el momento en que Amazon SQS recibe la solicitud `ReceiveMessage`. La ampliación del tiempo de espera de visibilidad no restablece el máximo de 12 horas.
> Además, es posible que no pueda establecer el tiempo de espera de un mensaje individual en 12 horas completas (por ejemplo, 43 200 segundos), ya que la solicitud `ReceiveMessage` inicia el temporizador. Por ejemplo, si recibe un mensaje e inmediatamente establece el máximo de 12 horas mediante el envío de una llamada `ChangeMessageVisibility` con `VisibilityTimeout` igual a 43 200 segundos, es probable que se produzca un error. No obstante, utilizar un valor de 43 195 segundos funcionará a menos que exista un retraso significativo entre la solicitud del mensaje a través de `ReceiveMessage` y la actualización del tiempo de espera de visibilidad. Si su consumidor necesita más de 12 horas, considere usar Step Functions.

Cambio del tiempo de espera de visibilidad de un mensaje

Cuando se recibe un mensaje de una cola y empieza a procesarse, el tiempo de espera de visibilidad de la cola puede ser insuficiente (por ejemplo, es posible que deba procesar y eliminar un mensaje). Puede especificar un nuevo valor de tiempo de espera mediante la acción `ChangeMessageVisibility` para reducir o ampliar la visibilidad de un mensaje.

Por ejemplo, si el tiempo de espera predeterminado de una cola es de 60 segundos, han transcurrido 15 segundos desde que recibió el mensaje y envía una llamada a `ChangeMessageVisibility` con `VisibilityTimeout` establecido en 10 segundos, los 10 segundos empiezan a contar a partir del momento en que se realiza la llamada a `ChangeMessageVisibility`. Por tanto, cualquier intento de cambiar el tiempo de espera de visibilidad o de eliminar el mensaje 10 segundos después de cambiar inicialmente el tiempo de espera de visibilidad (un total de 25 segundos) podría dar lugar a un error.

> **Note**
>
> El nuevo periodo de tiempo de espera entra en vigor desde el momento en que llama a la acción `ChangeMessageVisibility`. Además, el nuevo periodo de tiempo de espera se

aplica únicamente a la recepción concreta del mensaje. `ChangeMessageVisibility` no afecta al tiempo de espera de recepciones posteriores del mensaje ni a colas posteriores.

Finalización del tiempo de espera de visibilidad de un mensaje

Cuando recibe un mensaje de una cola, puede que realmente no desee procesarlo y eliminarlo. Amazon SQS le permite terminar el tiempo de espera de visibilidad de un mensaje concreto. De este modo, el mensaje será visible de inmediato para otros componentes del sistema y estará disponible para su procesamiento.

Para terminar el tiempo de espera de visibilidad de un mensaje después de llamar a `ReceiveMessage`, llame a `ChangeMessageVisibility` con `VisibilityTimeout` establecido en 0 segundos.

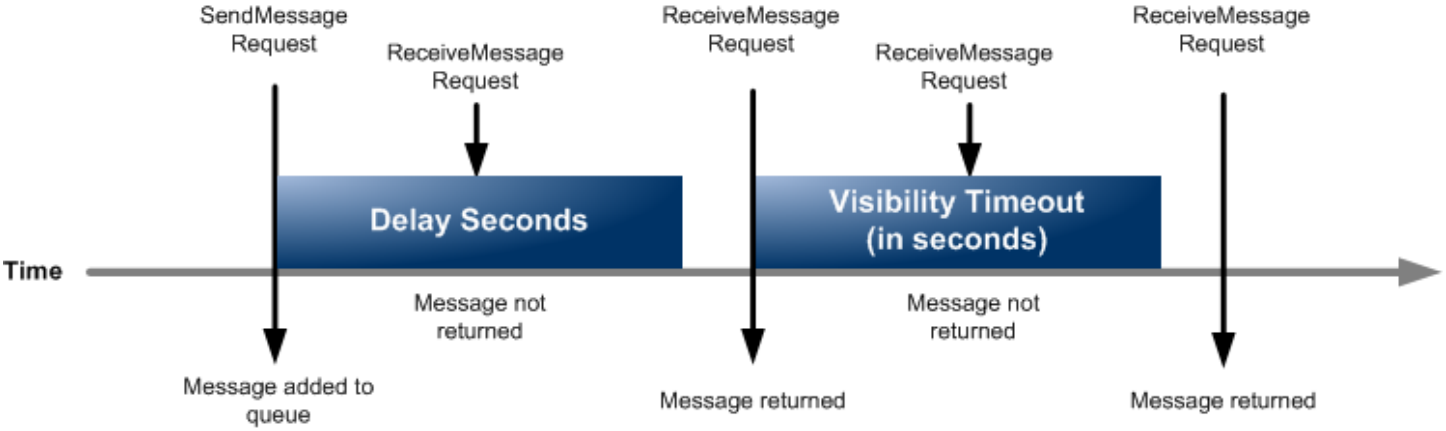
Colas con retraso de Amazon SQS

Las colas con retraso le permiten posponer la entrega de nuevos mensajes a los consumidores durante un número de segundos, por ejemplo, cuando su aplicación consumidora necesita tiempo adicional para procesar los mensajes. Si crea una cola con retraso, los mensajes que envíe a la cola permanecerán invisible para los consumidores mientras dure el período de retraso. El retraso predeterminado (mínimo) de una cola es de 0 segundos. El valor máximo es de 15 minutos. Para obtener información sobre la configuración las colas con retraso mediante la consola, consulte Configuración de los parámetros de cola mediante la consola Amazon SQS.

> **Note**
>
> En las colas estándar, la configuración de retraso por cola no es retroactiva, es decir, el cambio de la configuración no afecta al retraso de los mensajes que ya están en la cola. En las colas FIFO, la configuración de retraso por cola es retroactiva, es decir, el cambio de la configuración afecta al retraso de los mensajes que ya están en la cola.

Las colas con retraso son similares a los tiempos de espera de visibilidad porque ambos hacen que los mensajes no estén disponibles para los consumidores durante un periodo de tiempo determinado. La diferencia es que para las colas con retraso un mensaje está oculto cuando es la primera vez que se añade a la cola, mientras que para los tiempos de espera de visibilidad un

mensaje está oculto solo después de que se consuma un mensaje de la cola. El siguiente diagrama ilustra la relación existente entre las colas con retraso y los tiempos de espera de visibilidad.



Si desea establecer los segundos de retraso en mensajes específicos en lugar de en toda la cola, utilice temporizadores de mensajes para permitir que Amazon SQS use el valor `DelaySeconds` del temporizador de mensajes en lugar del valor `DelaySeconds` de la cola con retraso.

Colas temporales de Amazon SQS

Las colas temporales le ayudan a ahorrar tiempo de desarrollo y costes de implementación al utilizar patrones de mensajes comunes como, por ejemplo, solicitud-respuesta. Puede utilizar el Cliente de colas temporales para crear colas temporales de alto rendimiento, rentables y administradas por la aplicación.

El cliente asigna automáticamente varias colas temporales (colas administradas por la aplicación creadas bajo demanda para un proceso concreto) a una única cola de Amazon SQS. De este modo, la aplicación puede realizar menos llamadas y tiene un rendimiento superior cuando el tráfico de cada cola temporal es bajo. Cuando una cola temporal ya no está en uso, el cliente la limpia automáticamente, incluso aunque algunos procesos que utilicen el cliente no se hayan cerrado correctamente.

Estas son las ventajas que brindan las colas temporales:

- Actúan como canales de comunicación ligeros para procesos o subprocesos específicos.
- Se pueden crear y eliminar sin generar costos adicionales.
- Son compatibles con las colas estáticas (normales) de Amazon SQS a través de una API. Esto significa que el código existente que envía y recibe los mensajes puede enviar mensajes a colas virtuales y recibir mensajes de estas.

Temas

Colas virtuales

Las colas virtuales son estructuras de datos locales que el Cliente de colas temporales crea. Las colas virtuales permiten combinar varios destinos con poco tráfico en una única cola de Amazon SQS. Para obtener las prácticas recomendadas, consulte Evite reutilizar el mismo ID de grupo de mensajes con colas virtuales.

> **Note**
>
> - Cuando se crea una cola virtual, solo se crean estructuras de datos temporales en las que los consumidores reciben mensajes. Como las colas virtuales no hacen llamadas de API a Amazon SQS, no generan costo alguno.
> - Las cuotas de TPS se aplican a todas las colas virtuales en una sola cola de host. Para obtener más información, consulte Cuotas de mensajes de Amazon SQS.

La clase de encapsulamiento `AmazonSQSVirtualQueuesClient` proporciona compatibilidad con atributos relacionados con las colas virtuales. Para crear una cola virtual, debe llamar a la acción `CreateQueue` de la API con el atributo `HostQueueURL`. Este atributo especifica la cola existente que hospeda las colas virtuales.

La dirección URL de una cola virtual tiene el siguiente formato.

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue#MyVirtualQueueName
```

Cuando un productor llama a la acción `SendMessage` o `SendMessageBatch` de la API en una dirección URL de una cola virtual, el Cliente de colas temporales hace lo siguiente:

1. Extrae el nombre de la cola virtual.
2. Lo adjunta como un atributo más del mensaje.
3. Envía el mensaje a la cola del host.

Mientras el productor envía mensajes, un subproceso en segundo plano sondea la cola del host y envía los mensajes recibidos a las colas virtuales en función de los atributos del mensaje correspondiente.

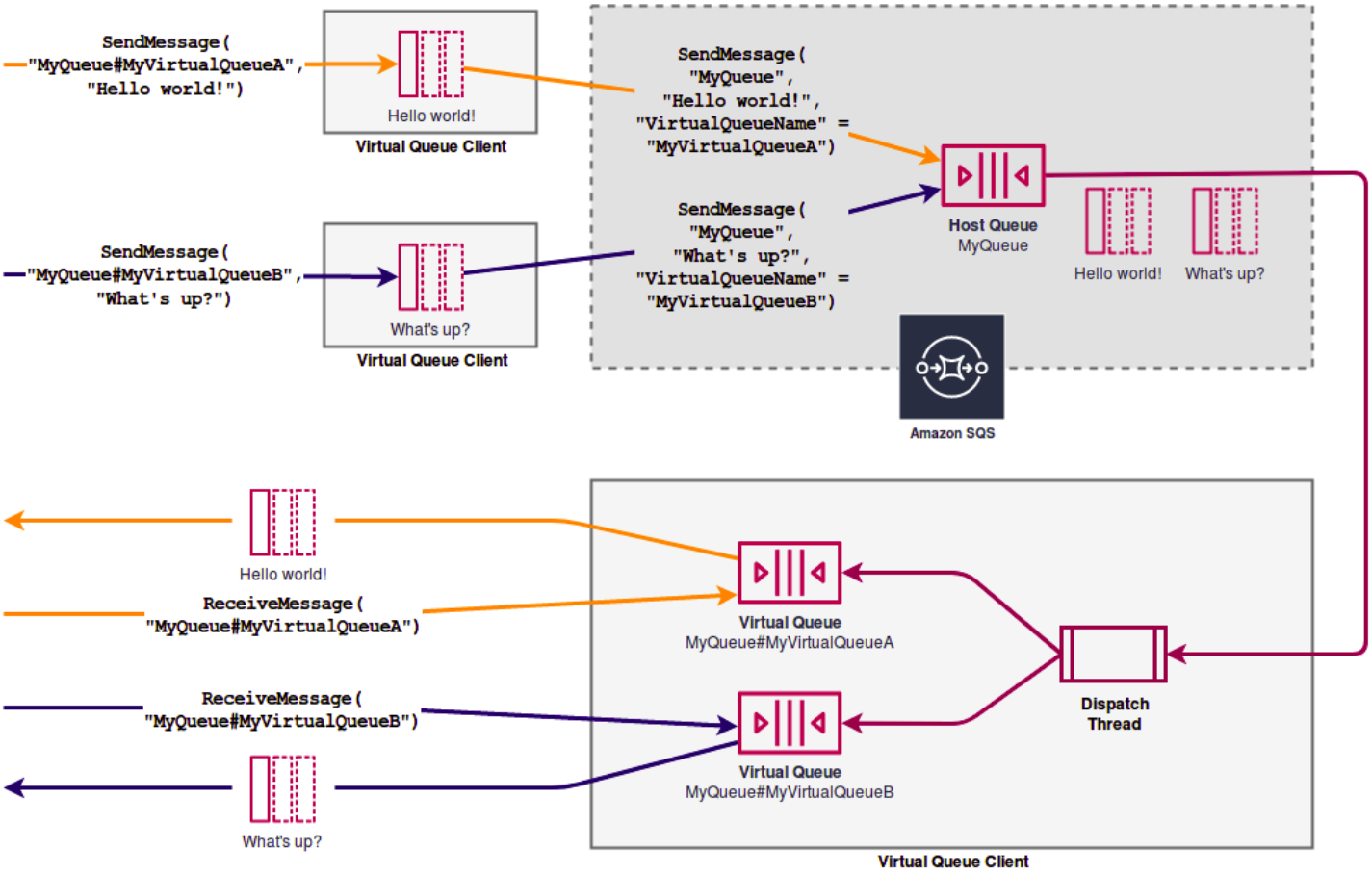
Mientras el consumidor llama a la acción `ReceiveMessage` de la API en una dirección URL de una cola virtual, el Cliente de colas temporales bloquea la llamada de este cliente localmente hasta que el subproceso en segundo plano envíe un mensaje a la cola virtual. (Este proceso es similar a la recopilación previa de mensajes del Cliente asíncrono en búfer: una sola acción de la API puede proporcionar mensajes en hasta 10 colas virtuales). Al eliminar una cola virtual, se elimina cualquier recurso del lado del cliente sin necesidad de llamar al propio servicio Amazon SQS.

La clase `AmazonSQSTemporaryQueuesClient` convierte automáticamente todas las colas que crea en colas temporales. También crea colas de host de forma automática con los mismos atributos, bajo demanda. Los nombres de estas colas comparten un prefijo común que se puede configurar (de forma predeterminada, `__RequesterClientQueues__`) y que las identifica como colas temporales. De este modo, el cliente puede actuar como un sustituto inmediato que optimiza el código existente, encargado de crear y eliminar las colas. El cliente también incluye las interfaces `AmazonSQSRequester` y `AmazonSQSResponder`, que posibilitan la comunicación bidireccional entre las colas.

Patrón de mensajes de respuesta a solicitudes (colas virtuales)

El caso de uso más común de las colas temporales es el patrón de mensajes de respuesta a solicitudes, donde un solicitante crea una cola temporal para recibir cada mensaje de respuesta. Para no tener que crear una cola de Amazon SQS por cada mensaje de respuesta, el Cliente de colas temporales permite crear y eliminar varias colas temporales sin necesidad de hacer llamadas a la API de Amazon SQS. Para obtener más información, consulte Implementación de sistemas de solicitud-respuesta.

En el siguiente diagrama, se muestra una configuración frecuente con este patrón.



Situación de ejemplo: procesamiento de una solicitud de inicio de sesión

En el siguiente ejemplo, se muestra cómo puede utilizar las interfaces `AmazonSQSRequester` y `AmazonSQSResponder` para procesar una solicitud de inicio de sesión del usuario.

En el lado del cliente

```
public class LoginClient {

    // Specify the Amazon SQS queue to which to send requests.
    private final String requestQueueUrl;

    // Use the AmazonSQSRequester interface to create
    // a temporary queue for each response.
    private final AmazonSQSRequester sqsRequester =
            AmazonSQSRequesterClientBuilder.defaultClient();

    LoginClient(String requestQueueUrl) {
        this.requestQueueUrl = requestQueueUrl;
```

```
    }

    // Send a login request.
    public String login(String body) throws TimeoutException {
        SendMessageRequest request = new SendMessageRequest()
                .withMessageBody(body)
                .withQueueUrl(requestQueueUrl);

        // If no response is received, in 20 seconds,
        // trigger the TimeoutException.
        Message reply = sqsRequester.sendMessageAndGetResponse(request,
                20, TimeUnit.SECONDS);

        return reply.getBody();
    }
}
```

Al enviar una solicitud de inicio de sesión, ocurre lo siguiente:

1. Se crea una tabla temporal.
2. Se adjunta la URL de la cola temporal al mensaje como un atributo.
3. Se envía el mensaje.
4. Se recibe una respuesta de la cola temporal.
5. Se elimina la cola temporal.
6. Se devuelve la respuesta.

En el lado del servidor

En el siguiente ejemplo se presupone que, en el momento de la construcción, se crea un subproceso que sondea la cola y llama al método `handleLoginRequest()` en cada mensaje. Además, se presupone que se utiliza el método `doLogin()`.

```
public class LoginServer {

    // Specify the Amazon SQS queue to poll for login requests.
    private final String requestQueueUrl;

    // Use the AmazonSQSResponder interface to take care
    // of sending responses to the correct response destination.
    private final AmazonSQSResponder sqsResponder =
```

```
            AmazonSQSResponderClientBuilder.defaultClient();

    LoginServer(String requestQueueUrl) {
        this.requestQueueUrl = requestQueueUrl;
    }

    // Process login requests from the client.
    public void handleLoginRequest(Message message) {

        // Process the login and return a serialized result.
        String response = doLogin(message.getBody());

        // Extract the URL of the temporary queue from the message attribute
        // and send the response to the temporary queue.
        sqsResponder.sendResponseMessage(MessageContent.fromMessage(message),
                new MessageContent(response));
    }
}
```

Limpieza de colas

Para asegurarse de que Amazon SQS recupera los recursos en memoria utilizados por las colas virtuales, cuando la aplicación ya no necesite el Cliente de colas temporales, debería llamar al método `shutdown()`. También se puede utilizar el método `shutdown()` de la interfaz `AmazonSQSRequester`.

El Cliente de colas temporales también cuenta con un mecanismo para eliminar las colas huérfanas del host. En cada una de las colas que recibe llamadas de la API durante un determinado periodo de tiempo (de forma predeterminada, cinco minutos), el cliente utiliza la acción `TagQueue` de la API para etiquetar las colas que permanecen en uso.

> **Note**
>
> Cualquier acción de la API que se realice en una cola marcará dicha cola como no inactiva, incluidas las acciones `ReceiveMessage` que no devuelven mensajes.

El subproceso en segundo plano utiliza las acciones `ListQueues` y `ListTags` de la API para comprobar todas las colas con el prefijo configurado y eliminar las colas que no se han etiquetado en los últimos cinco minutos. De esta forma, si un cliente no se cierra de forma limpia, el resto de los

clientes activos limpiarán después. Para reducir las cargas de trabajo duplicadas, todos los clientes con el mismo prefijo se comunican a través de una cola de trabajo interna y compartida.

Temporizadores de mensajes de Amazon SQS

Los temporizadores de mensajes le permiten especificar un período de invisibilidad inicial para un mensaje agregado a una cola. Por ejemplo, si envía un mensaje con un termporizador de 45 segundos, el mensaje no es visible para los consumidores durante los primeros 45 segundos en la cola. El retraso predeterminado (mínimo) de un mensaje es de 0 segundos. El valor máximo es de 15 minutos. Para obtener información sobre cómo enviar mensajes con temporizadores mediante la consola, consulte Enviar un mensaje.

> **Note**
>
> Las colas FIFO no admiten temporizadores en los mensajes individuales.

Para establecer un período de retraso en una cola entera, en lugar de en mensajes individuales, utilice colas de retraso. La configuración de un temporizador de mensajes para un mensaje individual anula cualquier valor `DelaySeconds` en una cola con retraso de Amazon SQS.

Acceso a Amazon EventBridge Pipes a través de la consola Amazon SQS

Amazon EventBridge Pipes conecta las fuentes con los objetivos. Los tubos están diseñados para la point-to-point integración entre las fuentes y los objetivos compatibles, y permiten realizar transformaciones y enriquecimientos avanzados. EventBridge Los tubos proporcionan una forma altamente escalable de conectar tu cola de Amazon SQS con AWS servicios como Step Functions, Amazon SQS y API Gateway, así como con aplicaciones de software como servicio (SaaS) de terceros, como Salesforce.

Para configurar una canalización, elija el origen, agregue filtros opcionales, defina el enriquecimiento opcional y elija el destino de los datos del evento.

En la página de detalles de una cola de Amazon SQS, puede ver las canalizaciones que utilizan ese clúster como origen. Desde allí, también puede hacer lo siguiente:

- Inicie la EventBridge consola para ver los detalles de la canalización.
- Inicie la EventBridge consola para crear una nueva tubería con la cola como fuente.

Para obtener más información sobre la configuración de una cola de Amazon SQS como fuente de canalización, consulte Amazon SQS cola como fuente en la Guía del usuario de Amazon. EventBridge Para obtener más información sobre EventBridge Pipes en general, consulte Pipes. EventBridge

Para acceder a EventBridge las canalizaciones de una cola de Amazon SQS determinada

1. En la consola de Amazon SQS, abra la página Colas.
2. Seleccione una cola.
3. En la página de detalles de la cola, seleccione la EventBridge pestaña Pipes.

   La pestaña EventBridge Tuberías incluye una lista de todas las tuberías actualmente configuradas para usar la cola seleccionada como fuente, que incluye:

   - nombre de la canalización
   - estado actual
   - destino de la canalización
   - cuándo se modificó la canalización por última vez
4. Consulte más detalles de la canalización o cree una nueva, si lo desea:

   - Para acceder a más detalles sobre una canalización:

     Elija el nombre de la canalización.

     Esto abre la página de detalles de Pipe de la EventBridge consola.
   - Para crear una nueva canalización:

     Elija Conectar la cola de Amazon SQS a la canalización.

     Esto abre la página Crear canalización de la EventBridge consola, con la cola Amazon SQS especificada como fuente de canalización. Para obtener más información, consulta Cómo crear una EventBridge tubería en la Guía del EventBridge usuario de Amazon.

> **Important**
>
> Una sola canalización lee un mensaje de una cola de Amazon SQS y, después, se elimina de la cola una vez procesado, independientemente de que el mensaje coincida o no con el filtro que puede configurar para esa canalización. Continúe con precaución al configurar varias canalizaciones para que utilicen la misma cola como origen.

Gestión de mensajes de Amazon SQS de gran tamaño con Extended Client Library y Amazon Simple Storage Service

Puede usar la biblioteca de clientes extendida de Amazon SQS para Java y la biblioteca de clientes extendida de Amazon SQS para Python para enviar mensajes de gran tamaño. Esto resulta especialmente útil cuando se consumen grandes cargas de mensajes, desde 256 KB hasta 2 GB. Ambas bibliotecas guardan la carga útil del mensaje en un bucket de Amazon Simple Storage Service y envían la referencia del objeto de Amazon S3 almacenado a la cola de Amazon SQS.

> **Note**
>
> Las bibliotecas de clientes ampliadas de Amazon SQS son compatibles con las colas estándar y FIFO.

Temas

- Administración de mensajes de Amazon SQS de gran tamaño mediante Java y Amazon S3
- Administración de mensajes de Amazon SQS de gran tamaño mediante Python y Amazon S3

Administración de mensajes de Amazon SQS de gran tamaño mediante Java y Amazon S3

Puede utilizar la biblioteca de clientes extendida Amazon SQS para Java y Amazon Simple Storage Service (Amazon S3) para gestionar mensajes grandes de Amazon Simple Queue Service (Amazon SQS). Esto resulta especialmente útil cuando se consumen grandes cargas de mensajes, desde 256 KB hasta 2 GB. La biblioteca guarda la carga útil del mensaje en un bucket de Amazon S3 y envía un mensaje con una referencia del objeto de Amazon S3 almacenado a una cola de Amazon SQS.

Puede utilizar la biblioteca de clientes ampliada de Amazon SQS para Java con el fin de hacer lo siguiente:

- Especificar si los mensajes se almacenan siempre en Amazon S3 o solo cuando su tamaño supera los 256 KB
- Enviar un mensaje que hace referencia a un único objeto de mensaje almacenado en un bucket de S3
- Recupera el objeto de mensaje de un bucket de Amazon S3
- Eliminar el objeto de mensaje de un bucket de Amazon S3

Requisitos previos

En el siguiente ejemplo, se utiliza el SDK de AWS Java. Para instalar y configurar el SDK, consulte Configurar el AWS SDK para Java en la Guía para AWS SDK for Java desarrolladores.

Antes de ejecutar el código de ejemplo, configure sus AWS credenciales. Para obtener más información, consulte Configurar AWS las credenciales y la región para el desarrollo en la Guía para AWS SDK for Java desarrolladores.

El SDK para Java y la biblioteca de clientes ampliada de Amazon SQS para Java requieren J2SE Development Kit 8.0 o una versión posterior.

> **Note**
>
> Puede utilizar la biblioteca de clientes ampliada de Amazon SQS para Java a fin de administrar los mensajes de Amazon SQS mediante Amazon S3 solo con AWS SDK for Java. No puede hacerlo con la AWS CLI consola de Amazon SQS, la API HTTP de Amazon SQS ni con ningún otro SDK. AWS

AWS Ejemplo de SDK for Java 2.x: uso de Amazon S3 para gestionar mensajes de Amazon SQS de gran tamaño

En el siguiente ejemplo de AWS SDK for Java 2.x, se crea un bucket de Amazon S3 con un nombre aleatorio y se añade una regla de ciclo de vida para eliminar objetos de forma permanente después de 14 días. También crea una cola denominada `MyQueue` y envía a la cola un mensaje al azar almacenado en un bucket de S3 y de más de 256 KB. Por último, el código recupera el mensaje, devuelve información sobre él y, a continuación, elimina el mensaje, la cola y el bucket.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

import com.amazon.sqs.javamessaging.AmazonSQSExtendedClient;
import com.amazon.sqs.javamessaging.ExtendedClientConfiguration;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;
import com.amazonaws.services.sqs.AmazonSQS;
import com.amazonaws.services.sqs.AmazonSQSClientBuilder;
import com.amazonaws.services.sqs.model.*;
import org.joda.time.DateTime;
import org.joda.time.format.DateTimeFormat;

import java.util.Arrays;
import java.util.List;
import java.util.UUID;

public class SQSExtendedClientExample {

// Create an Amazon S3 bucket with a random name.
private final static String S3_BUCKET_NAME = UUID.randomUUID() + "-"
        + DateTimeFormat.forPattern("yyMMdd-hhmmss").print(new DateTime());

public static void main(String[] args) {

    /*
     * Create a new instance of the builder with all defaults (credentials
     * and region) set automatically. For more information, see
     * Creating Service Clients in the AWS SDK for Java Developer Guide.
```

```
     */
    final AmazonS3 s3 = AmazonS3ClientBuilder.defaultClient();

    /*
     * Set the Amazon S3 bucket name, and then set a lifecycle rule on the
     * bucket to permanently delete objects 14 days after each object's
     * creation date.
     */
    final BucketLifecycleConfiguration.Rule expirationRule =
            new BucketLifecycleConfiguration.Rule();
    expirationRule.withExpirationInDays(14).withStatus("Enabled");
    final BucketLifecycleConfiguration lifecycleConfig =
            new BucketLifecycleConfiguration().withRules(expirationRule);

    // Create the bucket and allow message objects to be stored in the bucket.
    s3.createBucket(S3_BUCKET_NAME);
    s3.setBucketLifecycleConfiguration(S3_BUCKET_NAME, lifecycleConfig);
    System.out.println("Bucket created and configured.");

    /*
     * Set the Amazon SQS extended client configuration with large payload
     * support enabled.
     */
    final ExtendedClientConfiguration extendedClientConfig =
            new ExtendedClientConfiguration()
                    .withLargePayloadSupportEnabled(s3, S3_BUCKET_NAME);

    final AmazonSQS sqsExtended =
            new AmazonSQSExtendedClient(AmazonSQSClientBuilder
                    .defaultClient(), extendedClientConfig);

    /*
     * Create a long string of characters for the message object which will
     * be stored in the bucket.
     */
    int stringLength = 300000;
    char[] chars = new char[stringLength];
    Arrays.fill(chars, 'x');
    final String myLongString = new String(chars);

    // Create a message queue for this example.
    final String QueueName = "MyQueue" + UUID.randomUUID().toString();
    final CreateQueueRequest createQueueRequest =
            new CreateQueueRequest(QueueName);
```

```
    final String myQueueUrl = sqsExtended
            .createQueue(createQueueRequest).getQueueUrl();
    System.out.println("Queue created.");

    // Send the message.
    final SendMessageRequest myMessageRequest =
            new SendMessageRequest(myQueueUrl, myLongString);
    sqsExtended.sendMessage(myMessageRequest);
    System.out.println("Sent the message.");

    // Receive the message.
    final ReceiveMessageRequest receiveMessageRequest =
            new ReceiveMessageRequest(myQueueUrl);
    List<Message> messages = sqsExtended
            .receiveMessage(receiveMessageRequest).getMessages();

    // Print information about the message.
    for (Message message : messages) {
        System.out.println("\nMessage received.");
        System.out.println("  ID: " + message.getMessageId());
        System.out.println("  Receipt handle: " + message.getReceiptHandle());
        System.out.println("  Message body (first 5 characters): "
                + message.getBody().substring(0, 5));
    }

    // Delete the message, the queue, and the bucket.
    final String messageReceiptHandle = messages.get(0).getReceiptHandle();
    sqsExtended.deleteMessage(new DeleteMessageRequest(myQueueUrl,
            messageReceiptHandle));
    System.out.println("Deleted the message.");

    sqsExtended.deleteQueue(new DeleteQueueRequest(myQueueUrl));
    System.out.println("Deleted the queue.");

    deleteBucketAndAllContents(s3);
    System.out.println("Deleted the bucket.");
}

private static void deleteBucketAndAllContents(AmazonS3 client) {

    ObjectListing objectListing = client.listObjects(S3_BUCKET_NAME);

    while (true) {
        for (S3ObjectSummary objectSummary : objectListing
```

```
                .getObjectSummaries()) {
            client.deleteObject(S3_BUCKET_NAME, objectSummary.getKey());
        }

        if (objectListing.isTruncated()) {
            objectListing = client.listNextBatchOfObjects(objectListing);
        } else {
            break;
        }
    }

    final VersionListing list = client.listVersions(
            new ListVersionsRequest().withBucketName(S3_BUCKET_NAME));

    for (S3VersionSummary s : list.getVersionSummaries()) {
        client.deleteVersion(S3_BUCKET_NAME, s.getKey(), s.getVersionId());
    }

    client.deleteBucket(S3_BUCKET_NAME);
}
}
```

AWS Ejemplo de SDK for Java 2.x: uso de Amazon S3 para gestionar mensajes de Amazon SQS de gran tamaño

En el siguiente ejemplo de AWS SDK for Java 2.x, se crea un bucket de Amazon S3 con un nombre aleatorio y se añade una regla de ciclo de vida para eliminar objetos de forma permanente después de 14 días. También crea una cola denominada `MyQueue` y envía a la cola un mensaje al azar almacenado en un bucket de S3 y de más de 256 KB. Por último, el código recupera el mensaje, devuelve información sobre él y, a continuación, elimina el mensaje, la cola y el bucket.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
```

```
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

            import com.amazon.sqs.javamessaging.AmazonSQSExtendedClient;
import com.amazon.sqs.javamessaging.ExtendedClientConfiguration;
import org.joda.time.DateTime;
import org.joda.time.format.DateTimeFormat;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketLifecycleConfiguration;
import software.amazon.awssdk.services.s3.model.CreateBucketRequest;
import software.amazon.awssdk.services.s3.model.DeleteBucketRequest;
import software.amazon.awssdk.services.s3.model.DeleteObjectRequest;
import software.amazon.awssdk.services.s3.model.ExpirationStatus;
import software.amazon.awssdk.services.s3.model.LifecycleExpiration;
import software.amazon.awssdk.services.s3.model.LifecycleRule;
import software.amazon.awssdk.services.s3.model.LifecycleRuleFilter;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsRequest;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsResponse;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationRequest;
import software.amazon.awssdk.services.sqs.SqsClient;
import software.amazon.awssdk.services.sqs.model.CreateQueueRequest;
import software.amazon.awssdk.services.sqs.model.CreateQueueResponse;
import software.amazon.awssdk.services.sqs.model.DeleteMessageRequest;
import software.amazon.awssdk.services.sqs.model.DeleteQueueRequest;
import software.amazon.awssdk.services.sqs.model.Message;
import software.amazon.awssdk.services.sqs.model.ReceiveMessageRequest;
import software.amazon.awssdk.services.sqs.model.ReceiveMessageResponse;
import software.amazon.awssdk.services.sqs.model.SendMessageRequest;

import java.util.Arrays;
import java.util.List;
import java.util.UUID;


/**
 * Examples of using Amazon SQS Extended Client Library for Java 2.x
 *
 */
public class SqsExtendedClientExamples {
    // Create an Amazon S3 bucket with a random name.
```

```
    private final static String S3_BUCKET_NAME = UUID.randomUUID() + "-"
            + DateTimeFormat.forPattern("yyMMdd-hhmmss").print(new DateTime());

    public static void main(String[] args) {

        /*
         * Create a new instance of the builder with all defaults (credentials
         * and region) set automatically. For more information, see
         * Creating Service Clients in the AWS SDK for Java Developer Guide.
         */
        final S3Client s3 = S3Client.create();

        /*
         * Set the Amazon S3 bucket name, and then set a lifecycle rule on the
         * bucket to permanently delete objects 14 days after each object's
         * creation date.
         */
        final LifecycleRule lifeCycleRule = LifecycleRule.builder()
                .expiration(LifecycleExpiration.builder().days(14).build())
                .filter(LifecycleRuleFilter.builder().prefix("").build())
                .status(ExpirationStatus.ENABLED)
                .build();
        final BucketLifecycleConfiguration lifecycleConfig =
BucketLifecycleConfiguration.builder()
                .rules(lifeCycleRule)
                .build();

        // Create the bucket and configure it
        s3.createBucket(CreateBucketRequest.builder().bucket(S3_BUCKET_NAME).build());

s3.putBucketLifecycleConfiguration(PutBucketLifecycleConfigurationRequest.builder()
                .bucket(S3_BUCKET_NAME)
                .lifecycleConfiguration(lifecycleConfig)
                .build());
        System.out.println("Bucket created and configured.");

        // Set the Amazon SQS extended client configuration with large payload support
enabled
        final ExtendedClientConfiguration extendedClientConfig = new
ExtendedClientConfiguration().withPayloadSupportEnabled(s3, S3_BUCKET_NAME);

        final SqsClient sqsExtended = new
AmazonSQSExtendedClient(SqsClient.builder().build(), extendedClientConfig);
```

```
        // Create a long string of characters for the message object
        int stringLength = 300000;
        char[] chars = new char[stringLength];
        Arrays.fill(chars, 'x');
        final String myLongString = new String(chars);

        // Create a message queue for this example
        final String queueName = "MyQueue-" + UUID.randomUUID();
        final CreateQueueResponse createQueueResponse =
sqsExtended.createQueue(CreateQueueRequest.builder().queueName(queueName).build());
        final String myQueueUrl = createQueueResponse.queueUrl();
        System.out.println("Queue created.");

        // Send the message
        final SendMessageRequest sendMessageRequest = SendMessageRequest.builder()
                .queueUrl(myQueueUrl)
                .messageBody(myLongString)
                .build();
        sqsExtended.sendMessage(sendMessageRequest);
        System.out.println("Sent the message.");

        // Receive the message
        final ReceiveMessageResponse receiveMessageResponse =
sqsExtended.receiveMessage(ReceiveMessageRequest.builder().queueUrl(myQueueUrl).build());
        List<Message> messages = receiveMessageResponse.messages();

        // Print information about the message
        for (Message message : messages) {
            System.out.println("\nMessage received.");
            System.out.println("  ID: " + message.messageId());
            System.out.println("  Receipt handle: " + message.receiptHandle());
            System.out.println("  Message body (first 5 characters): " +
message.body().substring(0, 5));
        }

        // Delete the message, the queue, and the bucket
        final String messageReceiptHandle = messages.get(0).receiptHandle();

sqsExtended.deleteMessage(DeleteMessageRequest.builder().queueUrl(myQueueUrl).receiptHandle(me
        System.out.println("Deleted the message.");


sqsExtended.deleteQueue(DeleteQueueRequest.builder().queueUrl(myQueueUrl).build());
        System.out.println("Deleted the queue.");
```

```
        deleteBucketAndAllContents(s3);
        System.out.println("Deleted the bucket.");

    }

    private static void deleteBucketAndAllContents(S3Client client) {
        ListObjectsV2Response listObjectsResponse =
 client.listObjectsV2(ListObjectsV2Request.builder().bucket(S3_BUCKET_NAME).build());

        listObjectsResponse.contents().forEach(object -> {

 client.deleteObject(DeleteObjectRequest.builder().bucket(S3_BUCKET_NAME).key(object.key()).bui
        });

        ListObjectVersionsResponse listVersionsResponse =
 client.listObjectVersions(ListObjectVersionsRequest.builder().bucket(S3_BUCKET_NAME).build());

        listVersionsResponse.versions().forEach(version -> {

 client.deleteObject(DeleteObjectRequest.builder().bucket(S3_BUCKET_NAME).key(version.key()).ve
        });


 client.deleteBucket(DeleteBucketRequest.builder().bucket(S3_BUCKET_NAME).build());
    }
}
```

Puede utilizar Apache Maven para configurar y crear Amazon SQS Extended Client para su proyecto de Java o para crear el propio SDK. Especifique los módulos individuales del SDK que utiliza en su aplicación.

```
<properties>
    <aws-java-sdk.version>2.20.153</aws-java-sdk.version>
</properties>

<dependencies>
    <dependency>
      <groupId>software.amazon.awssdk</groupId>
      <artifactId>sqs</artifactId>
      <version>${aws-java-sdk.version}</version>
```

```
    </dependency>
    <dependency>
      <groupId>software.amazon.awssdk</groupId>
      <artifactId>s3</artifactId>
      <version>${aws-java-sdk.version}</version>
    </dependency>
    <dependency>
      <groupId>com.amazonaws</groupId>
      <artifactId>amazon-sqs-java-extended-client-lib</artifactId>
      <version>2.0.4</version>
    </dependency>

    <dependency>
      <groupId>joda-time</groupId>
      <artifactId>joda-time</artifactId>
      <version>2.12.6</version>
    </dependency>
</dependencies>
```

Administración de mensajes de Amazon SQS de gran tamaño mediante Python y Amazon S3

Puede utilizar la biblioteca de clientes extendida de Amazon Simple Queue Service para Python y Amazon Simple Storage Service para gestionar mensajes de Amazon SQS de gran tamaño. Esto resulta especialmente útil cuando se consumen grandes cargas de mensajes, desde 256 KB hasta 2 GB. La biblioteca guarda la carga útil del mensaje en un bucket de Amazon S3 y envía un mensaje con una referencia del objeto de Amazon S3 almacenado a una cola de Amazon SQS.

Puede usar la biblioteca cliente extendida para Python para hacer lo siguiente:

- Especifique si las cargas útiles se almacenan siempre en Amazon S3 o solo se almacenan en S3 cuando el tamaño de la carga supera los 256 KB
- Enviar un mensaje que haga referencia a un único objeto de mensaje almacenado en un bucket de Amazon S3
- Recupere el objeto de carga útil correspondiente de un bucket de Amazon S3
- Eliminar el objeto de carga útil correspondiente de un bucket de Amazon S3

Requisitos previos

Los siguientes son los requisitos previos para utilizar la biblioteca de clientes extendida Amazon SQS para Python:

- Una AWS cuenta con las credenciales necesarias. Para crear una AWS cuenta, vaya a la página de AWS inicio y, a continuación, seleccione Crear una AWS cuenta. Siga las instrucciones. Para obtener información sobre las credenciales, consulta Credenciales.
- Un AWS SDK: el ejemplo de esta página usa Boto3 del SDK de AWS Python. Para instalar y configurar el SDK, consulta la documentación del AWS SDK para Python en la Guía del desarrollador del AWS SDK para Python
- Python 3.x (o posterior) `ypip`.
- La biblioteca de clientes extendida Amazon SQS para Python, disponible en PyPI

> **Note**
>
> Puede utilizar la biblioteca de clientes extendida Amazon SQS para Python para gestionar los mensajes de Amazon SQS mediante Amazon S3 únicamente con el AWS SDK para Python. No puede hacerlo con la AWS CLI, la consola de Amazon SQS, la API HTTP de Amazon SQS ni con ningún otro SDK. AWS

Configuración del almacenamiento de mensajes

El cliente extendido de Amazon SQS utiliza los siguientes atributos de mensaje para configurar las opciones de almacenamiento de mensajes de Amazon S3:

- `large_payload_support`: el nombre del bucket de Amazon S3 para almacenar mensajes de gran tamaño.
- `always_through_s3`: Si`True`, entonces todos los mensajes se almacenan en Amazon S3. Si`False`, los mensajes de menos de 256 KB no se serializarán en el bucket s3. El valor predeterminado es `False`.
- `use_legacy_attribute`: Si`True`, todos los mensajes publicados utilizan el atributo de mensaje reservado heredado (`SQSLargePayloadSize`) en lugar del atributo de mensaje reservado actual (`ExtendedPayloadSize`).

Administración de mensajes de Amazon SQS de gran tamaño con la biblioteca de clientes extendida para Python

En el siguiente ejemplo, se crea un bucket de Amazon S3 con un nombre aleatorio. A continuación, crea una cola de Amazon SQS con el nombre `MyQueue` y envía un mensaje que está almacenado en un bucket de S3 y ocupa más de 256 KB a la cola. Por último, el código recupera el mensaje, devuelve información sobre él y, a continuación, elimina el mensaje, la cola y el bucket.

```
import boto3
import sqs_extended_client

#Set the Amazon SQS extended client configuration with large payload.
sqs_extended_client = boto3.client("sqs", region_name="us-east-1")
sqs_extended_client.large_payload_support = "S3_BUCKET_NAME"
sqs_extended_client.use_legacy_attribute = False


# Create an SQS message queue for this example. Then, extract the queue URL.
queue = sqs_extended_client.create_queue(
    QueueName = "MyQueue"
)
queue_url = sqs_extended_client.get_queue_url(
    QueueName = "MyQueue"
)['QueueUrl']


# Create the S3 bucket and allow message objects to be stored in the bucket.
sqs_extended_client.s3_client.create_bucket(Bucket=sqs_extended_client.large_payload_support)

# Sending a large message
small_message = "s"
large_message = small_message * 300000 # Shall cross the limit of 256 KB

send_message_response = sqs_extended_client.send_message(
    QueueUrl=queue_url,
    MessageBody=large_message
)
assert send_message_response['ResponseMetadata']['HTTPStatusCode'] == 200

# Receiving the large message
receive_message_response = sqs_extended_client.receive_message(
    QueueUrl=queue_url,
```

```
    MessageAttributeNames=['All']
)
assert receive_message_response['Messages'][0]['Body'] == large_message
receipt_handle = receive_message_response['Messages'][0]['ReceiptHandle']

# Deleting the large message
# Set to True for deleting the payload from S3
sqs_extended_client.delete_payload_from_s3 = True
delete_message_response = sqs_extended_client.delete_message(
    QueueUrl=queue_url,
    ReceiptHandle=receipt_handle
)

assert delete_message_response['ResponseMetadata']['HTTPStatusCode'] == 200

# Deleting the queue
delete_queue_response = sqs_extended_client.delete_queue(
    QueueUrl=queue_url
)

assert delete_queue_response['ResponseMetadata']['HTTPStatusCode'] == 200
```

Configuración de colas de Amazon SQS mediante la consola Amazon SQS

Utilice la consola de Amazon SQS para configurar y administrar las colas y características de Amazon Simple Queue Service (Amazon SQS). También puede usar la consola para configurar funciones como el cifrado del lado del servidor, asociar una cola de letra muerta a su cola o configurar un activador para invocar una función. AWS Lambda

Temas

Control de acceso basado en atributos para Amazon SQS

¿Qué es ABAC?

El control de acceso basado en atributos (ABAC) es un proceso de autorización que define los permisos en función de las etiquetas que se adjuntan a los usuarios y los recursos. AWS ABAC proporciona un control de acceso detallado y flexible basado en atributos y valores, reduce el riesgo de seguridad relacionado con las políticas basadas en roles reconfiguradas y centraliza la auditoría y la administración de las políticas de acceso. Para obtener más información sobre ABAC, consulte Qué es ABAC para AWS en la Guía del usuario de IAM.

Amazon SQS admite ABAC al permitirle controlar el acceso a las colas de Amazon SQS en función de las etiquetas y los alias asociados a una cola de Amazon SQS. Las claves de condición de etiqueta y alias que habilitan ABAC en Amazon SQS autorizan a las entidades principales de IAM a utilizar las colas de Amazon SQS sin necesidad de editar políticas ni administrar concesiones.

Con ABAC, puede utilizar etiquetas para configurar los permisos y las políticas de acceso de IAM para sus colas de Amazon SQS, lo que lo ayudará a escalar la administración de permisos. Puede crear una única política de permisos en IAM mediante etiquetas que agregue a cada rol empresarial, sin tener que actualizar la política cada vez que agregue un nuevo recurso. También puede asociar etiquetas a entidades principales de IAM para crear una política de ABAC. Puede diseñar políticas de ABAC para permitir las operaciones de Amazon SQS cuando la etiqueta del rol de usuario de IAM que está realizando la llamada coincida con la etiqueta de cola de Amazon SQS. Para obtener más información sobre el etiquetado AWS, consulte AWS Estrategias de etiquetado y. Etiquetas de asignación de costos de Amazon SQS

> **Note**
>
> ABAC para Amazon SQS está disponible actualmente en AWS todas las regiones comerciales en las que Amazon SQS está disponible, con las siguientes excepciones:
>
> - Asia-Pacífico (Hyderabad)
> - Asia-Pacífico (Melbourne)
> - Europa (España)
> - Europa (Zúrich)

¿Por qué debo utilizar ABAC en Amazon SQS?

He aquí algunos beneficios de utilizar ABAC en Amazon SQS:

- ABAC para Amazon SQS requiere menos políticas de permisos. No tiene que crear diferentes políticas para diferentes funciones de trabajo. Puede utilizar etiquetas de recursos y solicitudes que se apliquen a más de una cola, lo que reduce la sobrecarga operativa.
- Utilizar ABAC para escalar equipos rápidamente. Los permisos para nuevos recursos se conceden automáticamente en función de las etiquetas cuando los recursos se etiquetan adecuadamente durante su creación.

- Utilizar permisos en la entidad principal de IAM para restringir el acceso a los recursos. Puede crear etiquetas para la entidad principal de IAM y utilizarlas a fin de restringir el acceso a acciones específicas que coincidan con las etiquetas de la entidad principal de IAM. Esto lo ayuda a automatizar el proceso de concesión de permisos de solicitud.
- Realizar un seguimiento de quién accede a sus recursos. Puede determinar la identidad de una sesión si consulta los atributos de usuario en AWS CloudTrail.

Temas

Claves de condición de ABAC para Amazon SQS

Puede utilizar las siguientes claves de condición para controlar las acciones de las funciones:

Clave de condición de ABAC	Descripción	Tipo de política	Operaciones de Amazon SQS
leyes: ResourceTag	La etiqueta (clave y valor) en la cola de Amazon SQS coincide con la etiqueta (clave y valor) o el patrón de etiqueta en la política	Política de IAM únicamente	Operaciones de recursos de cola de Amazon SQS
leyes: RequestTag	La etiqueta (clave y valor) en las operaciones de recursos de la cola de Amazon SQS coincide con la etiqueta (clave y	Política de cola y políticas de IAM	TagQueue, UntagQueue, CreateQueue

Clave de condición de ABAC	Descripción	Tipo de política	Operaciones de Amazon SQS
	valor) o el patrón de etiqueta en la política		
leyes: TagKeys	Las claves de etiqueta de la solicitud coinciden con las claves de etiqueta de la política.	Política de cola y políticas de IAM	TagQueue, UntagQueue, CreateQueue

Etiquetado para el control de acceso en Amazon SQS

A continuación, se muestra un ejemplo de cómo utilizar etiquetas para el control de acceso. La política de IAM restringe a un usuario de IAM todas las acciones de Amazon SQS para todas las colas que incluyan una etiqueta de recurso con la clave environment y el valor production. Para obtener más información, consulte Control de acceso basado en atributos con etiquetas y Organizations AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessForProd",
      "Effect": "Deny",
      "Action": "sqs:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/environment": "prod"
        }
      }
    }
  ]
}
```

Creación de usuarios de IAM y colas de Amazon SQS

En los siguientes ejemplos se explica cómo crear una política ABAC para controlar el acceso a Amazon SQS mediante AWS Management Console y. AWS CloudFormation

Usando el AWS Management Console

Crear un usuario de IAM

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. Desde el panel de navegación izquierdo, elija Usuario.
3. Elija Agregar usuarios e introduzca un nombre en el cuadro de texto Nombre de usuario.
4. Seleccione (Clave de acceso: acceso mediante programación) y elija Siguiente: permisos.
5. Elija Siguiente:Etiquetas.
6. Agregue la clave de la etiqueta como `environment` y el valor de la etiqueta como `beta`.
7. Seleccione Siguiente: revisión y, después, elija Crear usuario.
8. Copie y guarde el ID de clave de acceso y la clave de acceso secreta en un lugar seguro.

Agregar permisos de usuario de IAM

1. Seleccione el usuario de IAM que ha creado.
2. Elija Agregar política insertada.
3. En la pestaña JSON, pegue la política siguiente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessForSameResTag",
      "Effect": "Allow",
      "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage"
      ],
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:ResourceTag/environment": "${aws:PrincipalTag/environment}"
        }
      }
    },
    {
      "Sid": "AllowAccessForSameReqTag",
      "Effect": "Allow",
      "Action": [
        "sqs:CreateQueue",
        "sqs:DeleteQueue",
        "sqs:SetQueueAttributes",
        "sqs:tagqueue"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "${aws:PrincipalTag/environment}"
        }
      }
    },
    {
      "Sid": "DenyAccessForProd",
      "Effect": "Deny",
      "Action": "sqs:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": "prod"
        }
      }
    }
  ]
}
```

4. Elija Revisar política.
5. Elija Crear política.

Usando AWS CloudFormation

Utilice la siguiente AWS CloudFormation plantilla de ejemplo para crear un usuario de IAM con una política en línea adjunta y una cola de Amazon SQS:

```
AWSTemplateFormatVersion: "2010-09-09"
Description: "CloudFormation template to create IAM user with custom inline policy"
Resources:
    IAMPolicy:
        Type: "AWS::IAM::Policy"
        Properties:
            PolicyDocument: |
                {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Sid": "AllowAccessForSameResTag",
                            "Effect": "Allow",
                            "Action": [
                                "sqs:SendMessage",
                                "sqs:ReceiveMessage",
                                "sqs:DeleteMessage"
                            ],
                            "Resource": "*",
                            "Condition": {
                                "StringEquals": {
                                    "aws:ResourceTag/environment": "${aws:PrincipalTag/
environment}"
                                }
                            }
                        },
                        {
                            "Sid": "AllowAccessForSameReqTag",
                            "Effect": "Allow",
                            "Action": [
                                "sqs:CreateQueue",
                                "sqs:DeleteQueue",
                                "sqs:SetQueueAttributes",
                                "sqs:tagqueue"
                            ],
                            "Resource": "*",
                            "Condition": {
                                "StringEquals": {
                                    "aws:RequestTag/environment": "${aws:PrincipalTag/
environment}"
                                }
                            }
                        },
```

```
                    {
                        "Sid": "DenyAccessForProd",
                        "Effect": "Deny",
                        "Action": "sqs:*",
                        "Resource": "*",
                        "Condition": {
                            "StringEquals": {
                                "aws:ResourceTag/stage": "prod"
                            }
                        }
                    }
                ]
            }

        Users:
          - "testUser"
        PolicyName: tagQueuePolicy

IAMUser:
    Type: "AWS::IAM::User"
    Properties:
        Path: "/"
        UserName: "testUser"
        Tags:
          -
            Key: "environment"
            Value: "beta"
```

Prueba del control de acceso basado en atributos

En los siguientes ejemplos se muestra cómo probar el control de acceso basado en atributos en Amazon SQS.

Cree una cola con la clave de etiqueta establecida a environment y el valor de etiqueta establecido a prod

Ejecute este comando AWS CLI para probar la creación de la cola con la clave de etiqueta establecida en environment y el valor de la etiqueta establecido en prod. Si no tiene AWS CLI, puede descargarla y configurarla para su máquina.

```
aws sqs create-queue --queue-name prodQueue –region us-east-1 –tags "environment=prod"
```

Recibe un error `AccessDenied` del punto de conexión de Amazon SQS:

```
An error occurred (AccessDenied) when calling the CreateQueue operation: Access to the
 resource <queueUrl> is denied.
```

Esto se debe a que el valor de etiqueta en el usuario de IAM no coincide con la etiqueta pasada en la llamada a la API `CreateQueue`. Recuerde que aplicamos una etiqueta al usuario de IAM con la clave establecida a `environment` y el valor establecido a `beta`.

Cree una cola con la clave de etiqueta establecida a environment y el valor de etiqueta establecido a beta

Ejecute este comando de la CLI para probar la creación de una cola con la clave de etiqueta establecida a `environment` y el valor de etiqueta establecido a `beta`.

```
aws sqs create-queue --queue-name betaQueue –region us-east-1 –tags "environment=beta"
```

Recibe un mensaje que confirma la creación correcta de la cola, similar al que se muestra a continuación.

```
{
"QueueUrl": "<queueUrl>“
}
```

Envío de un mensaje a una cola

Ejecute este comando de la CLI para probar el envío de un mensaje a una cola.

```
aws sqs send-message --queue-url <queueUrl> --message-body testMessage
```

La respuesta muestra una entrega correcta del mensaje a la cola de Amazon SQS. El permiso de usuario de IAM le permite enviar un mensaje a una cola que tenga una etiqueta `beta`. La respuesta incluye `MD5OfMessageBody` y `MessageId` con el mensaje.

```
{
"MD5OfMessageBody": "<MD5OfMessageBody>",
"MessageId": "<MessageId>"
}
```

Configuración de los parámetros de cola mediante la consola Amazon SQS

Al crear o editar una cola, puede configurar los siguientes parámetros:

- Tiempo de espera de visibilidad: el tiempo durante el que un mensaje recibido de una cola (por un consumidor) no estará visible para los demás consumidores de mensajes. Para obtener más información, consulte Tiempo de espera de visibilidad.

  > **Note**
  >
  > Utilizar la consola para configurar el tiempo de espera de visibilidad configura el valor del tiempo de espera para todos los mensajes de la cola. Para configurar el tiempo de espera de uno o varios mensajes, debes usar uno de los SDK. AWS

- Periodo de retención del mensaje: el tiempo que Amazon SQS retiene los mensajes que permanecen en la cola. De forma predeterminada, la cola retiene los mensajes durante cuatro días. Puede configurar una cola para que conserve los mensajes durante un máximo de 14 días. Para obtener más información, consulte Periodo de retención de mensajes.
- Retraso de entrega: el tiempo que Amazon SQS tardará en entregar un mensaje que se agregue a la cola. Para obtener más información, consulte Retraso de entrega.
- Tamaño máximo del mensaje: el tamaño máximo de mensaje para esta cola. Para obtener más información, consulte Tamaño máximo de los mensajes.
- Tiempo de espera de recepción del mensaje: el tiempo máximo que Amazon SQS espera a que los mensajes estén disponibles después de que la cola reciba una solicitud de recepción. Para obtener más información, consulte Sondeos cortos y largos de Amazon SQS.
- Activar la desduplicación basada en el contenido: Amazon SQS puede crear automáticamente identificadores de desduplicación basados en el cuerpo del mensaje. Para obtener más información, consulte Introducción a las colas FIFO en Amazon SQS.
- Habilitar FIFO de alto rendimiento: se utiliza para permitir un alto rendimiento de los mensajes en la cola. Al elegir esta opción, las opciones relacionadas (Ámbito de desduplicación y Límite de rendimiento FIFO) cambian a la configuración necesaria para habilitar un alto rendimiento para las colas FIFO. Para obtener más información, consulte Alto rendimiento de las colas FIFO en Amazon SQS y Cuotas de mensajes de Amazon SQS.

- Política de permiso de redireccionamiento: define qué colas de origen pueden utilizar esta cola como cola de mensajes fallidos. Para obtener más información, consulte Uso de colas de letra muerta en Amazon SQS .

Configuración de los parámetros de una cola existente (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas). Elija una cola y, a continuación, Editar.
3. Desplácese a la sección Configuración.
4. En Tiempo de espera de visibilidad, introduzca la duración y las unidades. El intervalo es de 0 segundos a 12 horas. El valor de predeterminado es de 30 segundos.
5. En Periodo de retención del mensaje, introduzca la duración y las unidades. El intervalo va de 1 minuto a 14 días. El valor predeterminado es 4 días.
6. Para una cola estándar, introduzca un valor en Tiempo de espera de recepción del mensaje. El intervalo es de 0 a 20 segundos. El valor predeterminado es 0 segundos, lo que establece un sondeo corto. Cualquier valor distinto de cero establece un sondeo largo.
7. En Retraso de entrega, introduzca la duración y las unidades. El intervalo es de 0 segundos a 15 minutos. El valor predeterminado es 0 segundos.
8. En Tamaño máximo del mensaje, introduzca un valor. El intervalo es de 1 KB a 256 KB. El valor predeterminado es 256 KB.
9. Para una cola FIFO, elija Habilitar la desduplicación basada en el contenido para activar la desduplicación basada en el contenido. La configuración predeterminada está desactivada.
10. (Opcional) Para que una cola FIFO permita un mayor rendimiento en el envío y recepción de mensajes en la cola, elija Habilitar FIFO de alto rendimiento.

    Al elegir esta opción, las opciones relacionadas (Ámbito de desduplicación y Límite de rendimiento FIFO) cambian a la configuración necesaria para habilitar un alto rendimiento para las colas FIFO. Si cambia alguna de las configuraciones necesarias para utilizar FIFO de alto rendimiento, el rendimiento normal estará en vigor para la cola y la desduplicación se producirá según lo especificado. Para obtener más información, consulte Alto rendimiento de las colas FIFO en Amazon SQS y Cuotas de mensajes de Amazon SQS.
11. En Política de permiso de redireccionamiento, elija Habilitado. Seleccione una de las siguientes opciones: Permitir todo (predeterminada), Por cola o Denegar todo. Al elegir Por cola, especifique una lista de hasta diez colas de origen por nombre de recurso de Amazon (ARN).

12. Cuando termine de configurar los parámetros de la cola, elija Guardar.

Configuración de la política de acceso

Al editar una cola, puede configurar su política de acceso.

La política de acceso define las cuentas, los usuarios y los roles que pueden acceder a la cola. La política de acceso también define las acciones (como `SendMessage`, `ReceiveMessage` o`DeleteMessage`) a las que pueden acceder los usuarios. La política predeterminada permite que solo el propietario de la cola envíe y reciba mensajes.

Configuración de la política de acceso para una cola existente (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. Elija una cola y, a continuación, Editar.
4. Desplácese hasta la sección Política de acceso.
5. Edite las instrucciones de la política de acceso en el cuadro de entrada. Para obtener más información sobre las instrucciones de política de acceso, consulte Identity and Access Management en Amazon SQS.
6. Cuando termine de configurar la política de acceso, seleccione Guardar.

Configuración del cifrado del lado del servidor para una cola mediante claves de cifrado administradas por SQL

Además de la opción predeterminada de cifrado del servidor (SSE) administrado por Amazon SQS, el SSE administrado por Amazon SQS (SSE-SQS) le permite crear un cifrado del servidor administrado personalizado que utiliza claves de cifrado administradas por SQS para proteger los datos confidenciales enviados a través de las colas de mensajes. Con SSE-SQS, no necesita crear ni administrar claves de cifrado, ni modificar su código para cifrar sus datos. SSE-SQS le permite transmitir datos de forma segura y le ayuda a cumplir con los estrictos requisitos de cifrado y normativos sin costo adicional.

SSE-SQS protege los datos en reposo mediante el cifrado AES-256 (estándar de cifrado avanzado de 256 bits). SSE cifra los mensajes en cuanto Amazon SQS los recibe. Amazon SQS almacena los mensajes de forma cifrada y los descifra solo cuando los envía a un consumidor autorizado.

> **Note**
>
> - La opción SSE predeterminada solo es efectiva cuando se crea una cola sin especificar atributos de cifrado.
> - Amazon SQS le permite desactivar todo el cifrado de las colas. Por lo tanto, desactivar KMS-SSE no activará automáticamente SQS-SSE. Si desea activar SQS-SSE después de desactivar KMS-SSE, deberá agregar un cambio de atributo en la solicitud.

Configuración del cifrado SSE-SQS en una cola (consola)

> **Note**
>
> Las colas nuevas que se creen mediante el punto de conexión HTTP (no TLS) no habilitarán el cifrado SSE-SQS de forma predeterminada. Una práctica recomendada en materia de seguridad consiste en crear colas de Amazon SQS mediante puntos de conexión HTTPS o Signature Version 4.

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. Elija una regla y, a continuación, elija Editar.
4. Expanda Cifrado.
5. Para Cifrado del servidor, elija Activado (predeterminado).

   > **Note**
   >
   > Con SSE activado, se rechazarán las solicitudes `SendMessage` y `ReceiveMessage` anónimas a la cola cifrada. Las prácticas recomendadas de seguridad de Amazon SQS desaconsejan utilizar solicitudes anónimas. Si desea enviar solicitudes anónimas a una cola de Amazon SQS, asegúrese de desactivar SSE.

6. Seleccione Clave de Amazon SQS (SSE-SQS). No hay ningún costo adicional por utilizar esta opción.
7. Elija Guardar.

Configuración del cifrado del lado del servidor para una cola mediante la consola Amazon SQS

Para proteger los datos de los mensajes de una cola, Amazon SQS tiene habilitado de forma predeterminada el cifrado del servidor (SSE) para todas las colas de nueva creación. Amazon SQS se integra con Amazon Web Services Key Management Service (Amazon Web Services KMS) para administrar las claves de KMS para el cifrado del servidor (SSE). Para obtener información acerca del uso de SSE, consulte Cifrado inactivo en Amazon SQS.

La clave de KMS que asigne a su cola debe tener una política de claves que incluya permisos para todas las entidades principales que estén autorizadas a utilizar la cola. Para obtener más información, consulte Administración de claves.

Si no es el propietario de la clave de KMS o si ha iniciado sesión con una cuenta que no tiene los permisos `kms:ListAliases` y `kms:DescribeKey`, no podrá ver la información sobre la clave de KMS en la consola de Amazon SQS. Pida al propietario de la clave de KMS que le conceda estos permisos. Para obtener más información, consulte Administración de claves.

Al crear o editar una cola, puede configurar SSE-KMS.

Configuración de SSE-KMS para una cola existente (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. Elija una regla y, a continuación, elija Editar.
4. Expanda Cifrado.
5. Para Cifrado del servidor, elija Activado (predeterminado).

> **Note**
>
> Con SSE activado, se rechazarán las solicitudes `SendMessage` y `ReceiveMessage` anónimas a la cola cifrada. Las prácticas recomendadas de seguridad de Amazon SQS desaconsejan utilizar solicitudes anónimas. Si desea enviar solicitudes anónimas a una cola de Amazon SQS, asegúrese de desactivar SSE.

6. Seleccione Clave de AWS Key Management Service (SSE-KMS).

La consola muestra la Descripción, la Cuenta y el ARN de la clave de KMS de la clave de KMS.

7. Especifique el ID de clave de KMS de la cola. Para obtener más información, consulte Términos clave.

   a. Elija la opción Elegir un alias de clave de KMS.

   b. La clave predeterminada es la clave de KMS administrada por Amazon Web Services para Amazon SQS. Para usar esta clave, selecciónela de la lista Clave de KMS.

   c. Para usar una clave de KMS personalizada de su cuenta de Amazon Web Services, selecciónela de la lista Clave de KMS. Para obtener instrucciones sobre la creación de claves de KMS personalizadas, consulte Creación de claves en la Guía para desarrolladores de Amazon Web Services Key Management Service.

   d. Para utilizar una clave de KMS personalizada que no esté en la lista o una clave de KMS personalizada de otra cuenta de Amazon Web Services, elija Introducir el alias de la clave de KMS e introduzca el Nombre de recurso de Amazon (ARN) de la clave de KMS.

8. (Opcional) Para Periodo de reutilización de la clave de datos, especifique un valor entre 1 minuto y 24 horas. El valor predeterminado es 5 minutos. Para obtener más información, consulte Descripción del período de reutilización de la clave de datos.

9. Cuando termine de configurar SSE-KMS, elija Guardar.

Configuración de etiquetas de asignación de costes para una cola mediante la consola Amazon SQS

Para ayudarlo a organizar e identificar sus colas de Amazon SQS, puede agregarles etiquetas de asignación de costos. Para obtener más información, consulte Etiquetas de asignación de costos de Amazon SQS.

En la página Detalles de una cola, la pestaña Etiquetado muestra las etiquetas de la cola.

Al crear o editar una cola, puede configurar etiquetas para ella.

Configuración de etiquetas para una cola existente (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. Elija una cola y, a continuación, Editar.

4. Desplácese hacia abajo hasta la sección Etiquetas.
5. Agregue, modifique o elimine etiquetas de cola:
   a. Para agregar una etiqueta, elija Agregar nueva etiqueta, introduzca una Clave y un Valor y, a continuación, elija Agregar nueva etiqueta.
   b. Para actualizar una etiqueta, cambie su Clave y Valor.
   c. Para eliminar una etiqueta, elija Eliminar junto a su par clave-valor.
6. Cuando termine de configurar las etiquetas, elija Guardar.

Suscripción de una cola a un tema de Amazon SNS mediante la consola Amazon SQS

Puede suscribir una o varias colas de Amazon SQS a un tema de Amazon Simple Notification Service (Amazon SNS). Cuando publica un mensaje en un tema, Amazon SNS lo envía a todas las colas suscritas. Amazon SQS administra la suscripción y todos los permisos necesarios. Para obtener más información sobre Amazon SNS, consulte ¿Qué es Amazon SNS? en la Guía para desarrolladores de Amazon Simple Notification Service.

Al suscribir una cola de Amazon SQS a un tema de Amazon SNS, Amazon SNS utiliza HTTPS para reenviar mensajes a Amazon SQS. Para obtener información sobre el uso de Amazon SNS con colas cifradas de Amazon SQS, consulte Configure los permisos de KMS para los servicios AWS.

> **Important**
>
> Amazon SQS admite como máximo 20 instrucciones por política de acceso. La suscripción a un tema de Amazon SNS agrega una instrucción de este tipo. Si se supera esta cantidad, se producirá un error en la entrega de la suscripción al tema.

Suscripción de una cola a un tema de SNS (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. En la lista de colas, elija la cola para suscribirse al tema de SNS.
4. En el menú Actions (Acciones), elija Subscribe to Amazon SNS topic (Suscribirse a tema de Amazon SNS).

5. Desde Especificar un tema de Amazon SNS disponible para este menú de cola, elija el tema de SNS para la cola.

   Si el tema SNS no aparece en el menú, elija Introducir ARN del tema de Amazon SNS y, a continuación, introduzca el nombre de recurso de Amazon (ARN) del tema.
6. Seleccione Guardar.
7. Para verificar el resultado de la suscripción, publique en el tema y, a continuación, consulte el mensaje que el tema envía a la cola. Para obtener más información, consulte Publicación de mensajes en Amazon SNS en la Guía para desarrolladores de Amazon Simple Notification Service.

Si la cola de Amazon SQS y el tema de SNS son diferentes Cuentas de AWS, el propietario del tema debe confirmar primero la suscripción. Para obtener más información, consulte Confirmar la suscripción en la Guía para desarrolladores de Amazon Simple Notification Service.

Para obtener información sobre la suscripción a un tema de SNS entre regiones, consulte Envío de mensajes de Amazon SNS a una cola o función de Amazon SQS en una región diferente AWS Lambda en la Guía para desarrolladores de Amazon Simple Notification Service

Configuración de una cola de Amazon SQS para activar una función AWS Lambda

Puede utilizar una AWS Lambda función para procesar los mensajes de una cola de Amazon SQS. Lambda sondea la cola e invoca su función de Lambda sincrónicamente con un evento que contiene mensajes de cola. Para permitir que su función tenga tiempo para procesar cada lote de registros, establezca el tiempo de espera de visibilidad de la cola de origen hasta al menos seis veces el tiempo de espera que configure en su función. El tiempo adicional permitirá a Lambda volver a intentar realizar los procesos en caso de que la función se vea limitada debido al procesamiento de un lote anterior.

Puede especificar otra cola para que actúe como una cola de mensajes fallidos de los mensajes que su función de Lambda no pueda procesar.

Una función de Lambda puede procesar elementos de varias colas (con un origen de eventos de Lambda para cada cola). Puede usar la misma cola con varias funciones de Lambda.

Si asocia una cola cifrada a una función de Lambda pero Lambda no sondea los mensajes, agregue el permiso `kms:Decrypt` a su rol de ejecución de Lambda.

Tenga en cuenta las siguientes restricciones:

- La cola y la función Lambda deben estar en la AWS misma región.
- Una cola cifrada que usa la clave predeterminada (clave de KMS AWS administrada para Amazon SQS) no puede invocar una función Lambda en otra. Cuenta de AWS

Para obtener información sobre la implementación de la función Lambda, consulte Uso AWS Lambda con Amazon SQS en AWS Lambda la Guía para desarrolladores.

Requisitos previos

Para configurar desencadenadores de funciones de Lambda, debe cumplir los siguientes requisitos:

- Si es un usuario, su rol de Amazon SQS debe incluir los siguientes permisos:
  - `lambda:CreateEventSourceMapping`
  - `lambda:ListEventSourceMappings`
  - `lambda:ListFunctions`
- El rol de ejecución de Lambda debe incluir los permisos siguientes:
  - `sqs:DeleteMessage`
  - `sqs:GetQueueAttributes`
  - `sqs:ReceiveMessage`
- Si asocia una cola cifrada a una función de Lambda, agregue el permiso `kms:Decrypt` a la función de ejecución de Lambda.

Para obtener más información, consulte Información general sobre la administración del acceso en Amazon SQS.

Configuración de una cola para desencadenar una función de Lambda (consola)

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. En la página Colas, elija la cola que desee configurar.
4. En la página de la cola, elija la pestaña Desencadenadores de Lambda.
5. En la página Desencadenadores de Lambda, elija un desencadenador de Lambda.

Si la lista no incluye el desencadenador de Lambda que necesita, elija Configurar desencadenador de función de Lambda. Introduzca el nombre de recurso de Amazon (ARN) de la función de Lambda o elija un recurso existente. A continuación, elija Guardar.

6. Seleccione Guardar. La consola guarda la configuración y muestra la página Detalles de la cola.

   En la página Detalles, la pestaña Desencadenadores de Lambda muestra la función de Lambda y su estado. Se tarda aproximadamente un minuto en asociar la función de Lambda a la cola.

7. Para verificar los resultados de la configuración, envíe un mensaje a la cola y, a continuación, consulte la función de Lambda desencadenada en la consola de Lambda.

Automatización de las notificaciones de AWS los servicios a Amazon SQS mediante Amazon EventBridge

Amazon EventBridge le permite automatizar AWS los servicios y responder a eventos del sistema, como problemas de disponibilidad de las aplicaciones o cambios en los recursos. Los eventos de AWS los servicios se entregan EventBridge prácticamente en tiempo real. Puede crear reglas sencillas para indicar qué eventos le resultan de interés, así como qué acciones automatizadas se van a realizar cuando un evento cumple una de las reglas.

EventBridge le permite establecer una variedad de objetivos, como el estándar Amazon SQS y las colas FIFO, que reciben eventos en formato JSON. Para obtener más información, consulta EventBridgelos objetivos de Amazon en la Guía del EventBridge usuario de Amazon.

Envío de un mensaje con atributos

En el caso de las colas estándar y FIFO, se pueden incluir metadatos estructurados (como marcas temporales, datos geoespaciales, firmas e identificadores) con los mensajes. Para obtener más información, consulte Atributos de mensajes de Amazon SQS.

Para enviar un mensaje con atributos a una cola mediante la consola Amazon SQS

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. En el panel de navegación, elija Queues (Colas).
3. En la página Colas, elija una cola.
4. Seleccione Enviar y recibir mensajes.

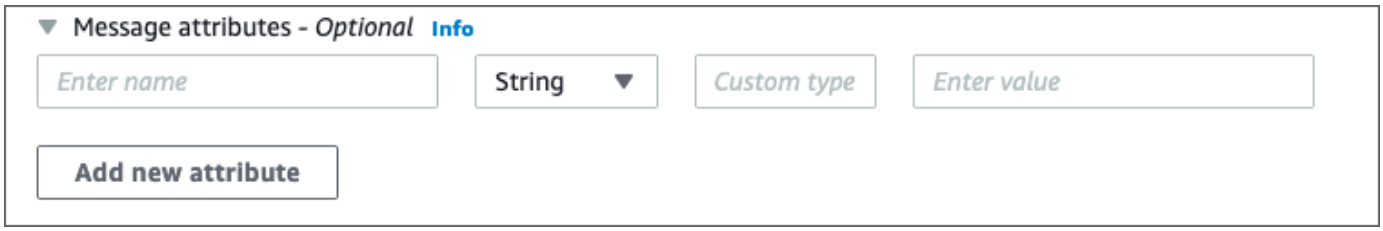
5. Introduzca los parámetros de los atributos del mensaje.

   a. En el cuadro de texto del nombre, introduzca un nombre único de 256 caracteres como máximo.

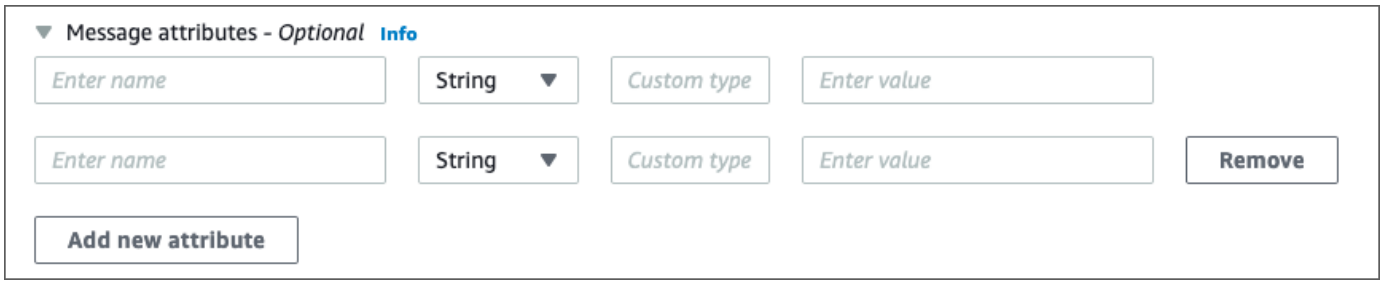
   b. Para el tipo de atributo, elija Cadena, Número o Binario.

   c. (Opcional) Introduzca un tipo de datos personalizado. Por ejemplo, puede agregar **`byte`**, **`int`** o **`float`** como tipos de datos personalizados para Número.

   d. En el cuadro de texto del valor, introduzca el valor de atributo del mensaje.



6. Para agregar otro atributo de mensaje, elija Agregar nuevo atributo.



7. Puede modificar los valores de atributo en cualquier momento antes de enviar el mensaje.

8. Para eliminar un atributo, elija Eliminar. Para eliminar el primer atributo, cierre Atributos del mensaje.

9. Cuando termine de agregar atributos al mensaje, elija Enviar mensaje. Se envía el mensaje y la consola muestra un mensaje de confirmación. Para ver información sobre los atributos del mensaje enviado, seleccione Ver detalles. Seleccione Listo para cerrar el cuadro de diálogo Detalles del mensaje.

Prácticas recomendadas para Amazon SQS

Estas prácticas recomendadas pueden ayudarlo a sacar el máximo partido de Amazon SQS.

Temas

Recomendaciones para las colas estándar y FIFO de Amazon SQS

Las siguientes prácticas recomendadas pueden ayudarlo a reducir costos y procesar los mensajes de forma eficaz mediante Amazon SQS.

Temas

Uso de los mensajes de Amazon SQS

Las siguientes directrices pueden ayudarlo a procesar los mensajes de forma eficaz mediante Amazon SQS.

Temas

Procesamiento de los mensajes a tiempo

La configuración del tiempo de espera de visibilidad depende de lo que tarde la aplicación en procesar y eliminar un mensaje. Por ejemplo, si la aplicación necesita 10 segundos para procesar un mensaje y establece el tiempo de espera de visibilidad en 15 minutos, debe esperar un tiempo relativamente largo para intentar procesar el mensaje de nuevo si el intento de procesamiento anterior produce un error. Asimismo, si la aplicación requiere 10 segundos para procesar un mensaje pero el usuario establece el tiempo de espera de visibilidad en solo 2 segundos, otro consumidor recibe un mensaje duplicado mientras que el consumidor original sigue trabajando en el mensaje.

Para asegurarse de que hay tiempo suficiente para procesar los mensajes, utilice una de las siguientes estrategias:

- Si sabe (o puede calcular razonablemente) el tiempo que se tarda en procesar un mensaje, amplíe el tiempo de espera de visibilidad del mensaje al tiempo máximo que se tarda en procesar y eliminar el mensaje. Para obtener más información, consulte Configuración del tiempo de espera de visibilidad.
- Si no sabe cuánto tiempo tarda en procesarse un mensaje, cree un latido para su proceso consumidor: especifique el tiempo de espera de visibilidad inicial (por ejemplo, dos minutos) y, a continuación, mientras su consumidor siga trabajando en el mensaje, continúe ampliando el tiempo de espera de visibilidad en dos minutos cada minuto.

> **Important**
>
> El tiempo máximo de visibilidad es de 12 horas desde el momento en que Amazon SQS recibe la solicitud `ReceiveMessage`. La ampliación del tiempo de espera de visibilidad no restablece el máximo de 12 horas.
> Además, es posible que no pueda establecer el tiempo de espera de un mensaje individual en 12 horas completas (por ejemplo, 43 200 segundos), ya que la solicitud `ReceiveMessage` inicia el temporizador. Por ejemplo, si recibe un mensaje e inmediatamente establece el máximo de 12 horas mediante el envío de una llamada `ChangeMessageVisibility` con `VisibilityTimeout` igual a 43 200 segundos, es probable que se produzca un error. No obstante, utilizar un valor de 43 195 segundos funcionará a menos que exista un retraso significativo entre la solicitud del mensaje a través de `ReceiveMessage` y la actualización del tiempo de espera de visibilidad. Si su consumidor necesita más de 12 horas, considere usar Step Functions.

Gestión de los errores de solicitud

Para gestionar los errores de solicitud, utilice una de las siguientes estrategias:

- Si utilizas un AWS SDK, ya tienes a tu disposición una lógica automática de reintentos y retrocesos. Para obtener más información, consulte Reintentos de error y retroceso exponencial en AWS en la Referencia general de Amazon Web Services.
- Si no utiliza las funciones del AWS SDK para volver a intentarlo y retrasar, espere una pausa (por ejemplo, 200 ms) antes de volver a intentar la ReceiveMessageacción si no recibe ningún mensaje, si se agota el tiempo de espera o si recibe un mensaje de error de Amazon SQS. Si quiere utilizar posteriormente `ReceiveMessage` con los mismos resultados, haga una pausa mayor (por ejemplo, 400 ms).

Configuración del sondeo largo

Cuando el tiempo de espera de la acción de la API `ReceiveMessage` es superior a 0, se está realizando un sondeo largo. El tiempo máximo de espera de sondeo es de 20 segundos. El sondeo largo ayuda a reducir el costo de uso de Amazon SQS al eliminar el número de respuestas vacías (cuando no hay ningún mensaje disponible para una solicitud `ReceiveMessage`) y las falsas respuestas vacías (cuando los mensajes están disponibles en la cola, pero no se incluyen en una respuesta). Para obtener más información, consulte Sondeos cortos y largos de Amazon SQS.

Para obtener un procesamiento óptimo de los mensajes, utilice las siguientes estrategias:

- En la mayoría de los casos, puede establecer el tiempo de espera de `ReceiveMessage` en 20 segundos. Si 20 segundos es demasiado tiempo para su aplicación, establezca un tiempo de espera de `ReceiveMessage` más corto (1 segundo como mínimo). Si no utiliza un AWS SDK para acceder a Amazon SQS o si configura un AWS SDK para que tenga un tiempo de espera más corto, es posible que tenga que modificar su cliente de Amazon SQS para permitir solicitudes más largas o utilizar un tiempo de espera más corto para sondeos prolongados.
- Si implementa el sondeo largo para varias colas, utilice un subproceso para cada cola en lugar de un solo subproceso para todas las colas. El uso de un único subproceso para cada cola permite que la aplicación procese los mensajes de cada una de las colas en cuanto estén disponibles, mientras que el uso de un único subproceso para sondear varias colas podría impedir que su aplicación procesara los mensajes disponibles en otras colas mientras la aplicación espera (hasta 20 segundos) para la cola que no tiene ningún mensaje disponible.

> **Important**
>
> Para evitar errores HTTP, asegúrese de que el tiempo de espera de respuesta HTTP de las solicitudes `ReceiveMessage` sea mayor que el parámetro `WaitTimeSeconds`. Para obtener más información, consulte. ReceiveMessage

Captura de mensajes problemáticos

Para capturar todos los mensajes que no se pueden procesar y recopilar CloudWatch métricas precisas, configura una cola de mensajes sin procesar.

- La política de redireccionamiento redirige los mensajes a una cola de mensajes fallidos después de que la cola de origen no puede procesar un mensaje un número especificado de veces.
- El uso de una cola de mensajes fallidos reduce el número de mensajes y la posibilidad de exponer el sistema a mensajes de píldoras venenosas (mensajes que se reciben pero no se pueden procesar).
- Incluir un mensaje sobre una píldora venenosa en una lista puede distorsionar la `ApproximateAgeOfOldestMessage` CloudWatch métrica al indicar una antigüedad incorrecta del mensaje sobre la píldora venenosa. Cuando se utiliza esta métrica, la configuración de una cola de mensajes fallidos ayuda a evitar falsas alarmas.

Configurar una retención de colas con letra muerta

En el caso de las colas estándar, la caducidad de un mensaje siempre se basa en su marca temporal original. Cuando un mensaje se mueve a una cola de mensajes fallidos, la marca temporal de la cola no se modifica. La métrica `ApproximateAgeOfOldestMessage` indica cuándo el mensaje pasó a la cola de mensajes fallidos, no cuándo se envió originalmente. Por ejemplo, supongamos que un mensaje pasa un día en la cola original antes de ser trasladado a una cola de mensajes fallidos. Si el periodo de retención de la cola de mensajes fallidos es de cuatro días, el mensaje se elimina de la cola de mensajes fallidos al cabo de tres días y `ApproximateAgeOfOldestMessage` es de tres días. Por lo tanto, se recomienda establecer siempre un periodo de retención de una cola de mensajes fallidos superior al periodo de retención de la cola original.

Para las colas FIFO, la marca temporal de entrada se restablece cuando el mensaje se mueve a una cola de mensajes fallidos. La métrica `ApproximateAgeOfOldestMessage` indica cuándo el mensaje ha pasado a la cola de mensajes fallidos. En el mismo ejemplo

anterior, el mensaje se elimina de la cola de mensajes fallidos al cabo de cuatro días y `ApproximateAgeOfOldestMessage` es de cuatro días.

Cómo evitar el procesamiento incoherente de los mensajes

Debido a que Amazon SQS es un sistema distribuido, es posible que un consumidor no reciba un mensaje aunque lo marque como entregado al regresar correctamente de una llamada de método de API `ReceiveMessage`. En este caso, Amazon SQS registra el mensaje como entregado al menos una vez, aunque el consumidor no lo haya recibido. Dado que no se realizan intentos adicionales de entregar mensajes en estas condiciones, no recomendamos establecer el número máximo de recepciones en 1 para una cola de mensajes fallidos.

Implementación de sistemas de solicitud-respuesta

Al implementar un sistema de solicitud-respuesta o de llamada a procedimiento remoto (RPC), tenga en cuenta las siguientes prácticas recomendadas:

- No cree colas de respuestas por mensaje. En su lugar, cree colas de respuesta al inicio, por productor, y utilice un atributo de mensaje de ID de correlación para asignar las respuestas a las solicitudes.
- No permita que sus productores compartan colas de respuestas. Esto puede provocar que un productor reciba mensajes de respuesta destinados a otro productor.

Para obtener más información acerca de cómo implementar el patrón de solicitud de respuesta utilizando el Cliente de colas temporales, consulte Patrón de mensajes de respuesta a solicitudes (colas virtuales).

Reducción de costos de Amazon SQS

Las siguientes prácticas recomendadas pueden ayudarle a reducir costos y aprovechar la posible reducción de costos adicionales y una respuesta casi instantánea.

Procesamiento por lotes de las acciones de los mensajes

Para reducir los costos, procese por lotes las acciones de los mensajes:

- Para enviar, recibir y eliminar mensajes, y para cambiar el tiempo de espera de visibilidad de varios mensajes con una sola acción, utilice las acciones de la API de procesamiento por lotes de Amazon SQS.

- Para combinar el almacenamiento en búfer del lado del cliente con el agrupamiento en lotes de solicitudes, utilice el sondeo largo junto con el cliente asincrónico en búfer incluido con la AWS SDK for Java.

> **Note**
>
> El cliente asincrónico con búfer de Amazon SQS no admite actualmente las colas FIFO.

Uso del modo de sondeo apropiado

- El sondeo largo le permite consumir mensajes de la cola de Amazon SQS tan pronto como estén disponibles.
  - Para reducir el costo derivado del uso de Amazon SQS y reducir el número de recepciones vacías en una cola vacía (respuestas a la acción `ReceiveMessage` que no devuelven ningún mensaje), habilite el sondeo largo. Para obtener más información, consulte Sondeo largo de Amazon SQS.
  - Para aumentar la eficacia cuando se sondean varios subprocesos con varias recepciones, reduzca el número de procesos.
  - En la mayoría de los casos, el sondeo largo es preferible al sondeo corto.
- El sondeo corto devuelve respuestas inmediatamente, incluso aunque la cola de Amazon SQS sondeada esté vacía.
  - Para satisfacer los requisitos de una aplicación que espera respuestas inmediatas a la solicitud `ReceiveMessage`, utilice el sondeo corto.
  - El sondeo corto se factura igual que el sondeo largo.

Cambio de una cola estándar a una cola FIFO de Amazon SQS

Si no establece el parámetro `DelaySeconds` en cada mensaje, puede cambiar a una cola FIFO si proporciona un ID de grupo de mensajes para cada mensaje enviado.

Para obtener más información, consulte Pasar de una cola estándar a una cola FIFO en Amazon SQS.

Recomendaciones adicionales para las colas FIFO de Amazon SQS

Las siguientes prácticas recomendadas pueden ayudarle a utilizar el ID de desduplicación de mensajes y el ID de grupo de mensajes de forma óptima. Para obtener más información, consulte las acciones `SendMessage` y `SendMessageBatch` en la Referencia de la API de Amazon Simple Queue Service.

Temas

Uso del ID de desduplicación de mensajes de Amazon SQS

El ID de desduplicación de mensajes es el token utilizado para la desduplicación de los mensajes enviados. Si un mensaje con un ID de desduplicación de mensajes concreto se envía correctamente, todos los mensajes enviados con el mismo ID de desduplicación de mensajes se aceptan correctamente pero no se entregan durante el intervalo de desduplicación de 5 minutos.

> **Note**
>
> Amazon SQS sigue realizando un seguimiento del ID de desduplicación del mensaje incluso después de haberlo recibido y eliminado.

Proporcionar el ID de desduplicación de mensajes

El productor debe proporcionar los valores de ID de desduplicación de mensajes para cada mensaje en los siguientes escenarios:

- Mensajes enviados con cuerpos idénticos que Amazon SQS debe tratar como únicos.
- Mensajes enviados con contenido idéntico pero atributos diferentes que Amazon SQS debe tratar como únicos.
- Mensajes enviados con contenido diferente (por ejemplo, los recuentos de reintentos están incluidos en el cuerpo del mensaje) que Amazon SQS debe tratar como duplicados.

Habilitación de la desduplicación para un sistema de un solo productor y un solo consumidor

Si tiene un único productor y un único consumidor y los mensajes son exclusivos porque se incluye un ID de mensaje específico de la aplicación en el cuerpo del mensaje, siga estas prácticas recomendadas:

- Habilite la desduplicación basada en el contenido para la cola (cada uno de sus mensajes tiene un cuerpo único). El productor puede omitir el ID de desduplicación de mensajes.
- Cuando la deduplicación basada en contenido está habilitada para una cola FIFO de Amazon SQS y se envía un mensaje con un ID de deduplicación, el ID de deduplicación `SendMessage` anula el ID de deduplicación basada en contenido generado.
- Aunque no es necesario que el consumidor proporcione un ID de intento de solicitud de recepción para cada solicitud, es recomendable hacerlo porque permite que las secuencias de reintento tras un error se ejecuten con mayor rapidez.
- Puede reintentar las solicitudes de envío o recepción, ya que no interfieren con la ordenación de los mensajes en las colas FIFO.

Diseño para situaciones de recuperación de interrupciones

El proceso de desduplicación en las colas FIFO está sujeto a limitación temporal. Al diseñar una aplicación, asegúrese de que tanto el productor como el consumidor puedan recuperarse en caso de que se produzca una interrupción del cliente o de la red.

- El productor debe estar al tanto del intervalo de desduplicación de la cola. Amazon SQS tiene un intervalo mínimo de desduplicación de cinco minutos. El reintento de solicitudes `SendMessage` después de que finalice el intervalo de desduplicación puede introducir mensajes duplicados en la cola. Por ejemplo, un dispositivo móvil en un automóvil envía mensajes cuyo orden es importante. Si el automóvil pierde la conectividad móvil durante un periodo de tiempo antes de recibir un reconocimiento, el reintento de la solicitud después de recuperar la conectividad móvil puede crear un duplicado.
- El consumidor debe tener un tiempo de espera de visibilidad que minimice el riesgo de no poder procesar los mensajes antes de que finalice el tiempo de espera de visibilidad. Para ampliar el tiempo de espera de visibilidad mientras se procesan los mensajes, llame a la acción `ChangeMessageVisibility`. Sin embargo, si el tiempo de espera de visibilidad finaliza, otro consumidor puede comenzar inmediatamente a procesar los mensajes, lo que hará que un

mensaje se procese varias veces. Para evitar esta situación, configure una cola de mensajes fallidos.

Uso de tiempos de espera de visibilidad

Para obtener un rendimiento óptimo, establece el tiempo de espera de visibilidad para que sea mayor que el tiempo de espera de lectura del AWS SDK. Hágalo cuando utilice la acción `ReceiveMessage` de la API tanto con sondeos cortos como con sondeos largos.

Uso del ID de grupo de mensajes de Amazon SQS

`MessageGroupId` es la etiqueta que especifica que un mensaje pertenece a un grupo de mensajes específico. Los mensajes que pertenecen al mismo grupo de mensajes se procesan siempre uno a uno, en un orden estricto relativo al grupo de mensajes (no obstante, los mensajes que pertenecen a grupos de mensajes diferentes podrían procesarse sin orden).

Intercalación de varios grupos de mensajes ordenados

Para intercalar varios grupos de mensajes ordenados dentro de una única cola FIFO, debe utilizar valores de ID de grupo de mensajes (por ejemplo, datos de sesiones para varios usuarios). En esta situación, varios consumidores pueden procesar la cola, pero los datos de sesión de cada usuario se procesan según el modelo FIFO.

> **Note**
>
> Cuando los mensajes que pertenecen a un determinado ID de grupo de mensajes son invisibles, ningún otro consumidor puede procesar los mensajes que tienen el mismo ID de grupo de mensajes.

Cómo evitar procesar los duplicados en un sistema de varios productores y consumidores

Para evitar el procesamiento de mensajes duplicados en un sistema que cuenta con varios productores y consumidores y donde el desempeño y la latencia son más importantes que la ordenación, el productor debería generar un ID de grupo de mensajes único para cada mensaje.

> **Note**
>
> En esta situación, se eliminan los duplicados. Sin embargo, la ordenación de los mensaje no se puede garantizar.
> Cualquier escenario que tenga varios productores y consumidores aumenta el riesgo de entregar accidentalmente un mensaje duplicado si un proceso de trabajo no procesa el mensaje durante el tiempo de espera de visibilidad y el mensaje está disponible para otro proceso de trabajo.

Evite tener una gran cantidad de tareas pendientes de mensajes con el mismo ID de grupo de mensajes

En el caso de las colas FIFO, puede haber un máximo de 20 000 mensajes en tránsito (recibidos de una cola por un consumidor, pero aún no eliminados de la cola). Si alcanza esta cuota, Amazon SQS no devuelve ningún mensaje de error. Una cola FIFO examina los primeros 20 000 mensajes para determinar los grupos de mensajes disponibles. Esto significa que si tiene mensajes pendientes en un único grupo de mensajes, no podrá consumir mensajes de otros grupos de mensajes que se hayan enviado a la cola posteriormente hasta que consuma correctamente los mensajes pendientes.

> **Note**
>
> Es posible que se acumulen tareas pendientes de mensajes que tengan el mismo ID de grupo de mensajes debido a que un consumidor no puede procesar correctamente un mensaje. Los problemas de procesamiento de mensajes pueden producirse debido a un problema con el contenido de un mensaje o debido a un problema técnico con el consumidor. Para alejar los mensajes que no se pueden procesar repetidamente y desbloquear el procesamiento de otros mensajes que tienen el mismo ID de grupo de mensajes, considere la posibilidad de configurar una política de cola de mensajes fallidos.

Evite reutilizar el mismo ID de grupo de mensajes con colas virtuales

Para evitar que los mensajes con el mismo ID de grupo de mensajes enviados a diferentes colas virtuales con la misma cola de host se bloqueen entre sí, intente reutilizar el mismo ID de grupo de mensajes con colas virtuales.

Uso del ID de intento de solicitud de recepción de Amazon SQS

El ID de intento de solicitud de recepción es el token utilizado para la desduplicación de llamadas `ReceiveMessage`.

Durante una interrupción de la red prolongada que ocasiona problemas de conectividad entre su SDK y Amazon SQS, se recomienda proporcionar el ID de intento de solicitud de recepción y volver a intentarlo con ese mismo ID de intento de solicitud de recepción si la operación del SDK produce un error.

Ejemplos de SDK de Java de Amazon SQS

Puede utilizarla AWS SDK for Java para crear aplicaciones Java que interactúen con Amazon Simple Queue Service (Amazon SQS) y otros servicios. AWS Para instalar y configurar el SDK, consulte Introducción en la Guía para desarrolladores de AWS SDK for Java 2.x .

Para ver ejemplos de operaciones básicas con colas de Amazon SQS, como la creación de una cola o el envío de un mensaje, consulte Uso de colas de mensajes de Amazon SQS en la Guía para desarrolladores de AWS SDK for Java 2.x .

En los ejemplos de este tema se demuestran características de Amazon SQS adicionales, como el cifrado del servidor (SSE), las etiquetas de asignación de costos y atributos de mensajes.

Temas

Uso del cifrado del lado del servidor con colas de Amazon SQS

Puede utilizarla AWS SDK for Java para añadir el cifrado del lado del servidor (SSE) a una cola de Amazon SQS. Cada cola utiliza una clave KMS AWS Key Management Service (AWS KMS) para generar las claves de cifrado de datos. En este ejemplo, se usa la clave de KMS AWS administrada para Amazon SQS. Para obtener más información sobre cómo utilizar SSE y el rol de la clave de KMS, consulte Cifrado inactivo en Amazon SQS.

Agregar SSE a una cola existente

Para activar el cifrado del servidor en una cola existente, utilice el método `SetQueueAttributes` para establecer el atributo `KmsMasterKeyId`.

El siguiente ejemplo de código establece la AWS KMS key clave de KMS AWS administrada para Amazon SQS. En el ejemplo también se establece el periodo de reutilización de AWS KMS key a 140 segundos.

Antes de ejecutar el código de ejemplo, asegúrese de haber establecido sus AWS credenciales. Para obtener más información, consulte Configurar AWS las credenciales y la región para el desarrollo en la Guía para AWS SDK for Java 2.x desarrolladores.

```
// Create an SqsClient for the specified Region.
SqsClient sqsClient = SqsClient.builder().region(Region.US_WEST_1).build();

// Get the URL of your queue.
String myQueueName = "my queue";
GetQueueUrlResponse getQueueUrlResponse =

 sqsClient.getQueueUrl(GetQueueUrlRequest.builder().queueName(myQueueName).build());
String queueUrl = getQueueUrlResponse.queueUrl();

// Create a hashmap for the attributes. Add the key alias and reuse period to the
 hashmap.
HashMap<QueueAttributeName, String> attributes = new HashMap<QueueAttributeName,
 String>();
final String kmsMasterKeyAlias = "alias/aws/sqs";  // the alias of the AWS managed KMS
 key for Amazon SQS.
attributes.put(QueueAttributeName.KMS_MASTER_KEY_ID, kmsMasterKeyAlias);
attributes.put(QueueAttributeName.KMS_DATA_KEY_REUSE_PERIOD_SECONDS, "140");

// Create the SetQueueAttributesRequest.
SetQueueAttributesRequest set_attrs_request = SetQueueAttributesRequest.builder()
          .queueUrl(queueUrl)
          .attributes(attributes)
          .build();

sqsClient.setQueueAttributes(set_attrs_request);
```

Desactivación de SSE para una cola

Para desactivar el cifrado del servidor para una cola existente, establezca el atributo `KmsMasterKeyId` a una cadena vacía mediante el método `SetQueueAttributes`.

> **Important**
>
> `null` no es un valor válido para `KmsMasterKeyId`.

Creación de una cola con SSE

Para activar SSE al crear la cola, agregue el atributo `KmsMasterKeyId` al método de la API `CreateQueue`.

En el siguiente ejemplo se crea una nueva cola con SSE activado. La cola utiliza la clave de KMS administrada por AWS para Amazon SQS. En el ejemplo también se establece el periodo de reutilización de AWS KMS key a 160 segundos.

Antes de ejecutar el código de ejemplo, asegúrese de haber establecido sus AWS credenciales. Para obtener más información, consulte Configurar AWS las credenciales y la región para el desarrollo en la Guía para AWS SDK for Java 2.x desarrolladores.

```
// Create an SqsClient for the specified Region.
SqsClient sqsClient = SqsClient.builder().region(Region.US_WEST_1).build();

// Create a hashmap for the attributes. Add the key alias and reuse period to the
 hashmap.
HashMap<QueueAttributeName, String> attributes = new HashMap<QueueAttributeName,
 String>();
final String kmsMasterKeyAlias = "alias/aws/sqs";  // the alias of the AWS managed KMS
 key for Amazon SQS.
attributes.put(QueueAttributeName.KMS_MASTER_KEY_ID, kmsMasterKeyAlias);
attributes.put(QueueAttributeName.KMS_DATA_KEY_REUSE_PERIOD_SECONDS, "140");

// Add the attributes to the CreateQueueRequest.
CreateQueueRequest createQueueRequest =
                CreateQueueRequest.builder()
                        .queueName(queueName)
                        .attributes(attributes)
                        .build();
sqsClient.createQueue(createQueueRequest);
```

Recuperación de atributos de SSE

Para obtener información sobre la recuperación de atributos de cola, consulte Ejemplos en la Referencia de la API de Amazon Simple Queue Service.

Para recuperar el ID de la clave de KMS o el periodo de reutilización de la clave de datos para una cola en concreto, ejecute el método `GetQueueAttributes` y recupere los valores `KmsMasterKeyId` y `KmsDataKeyReusePeriodSeconds`.

Configuración de etiquetas para una cola de Amazon SQS

Utilice etiquetas de asignación de costos para organizar e identificar sus colas de Amazon SQS. En los siguientes ejemplos se muestra cómo configurar etiquetas mediante la función AWS SDK for Java. Para obtener más información, consulte Etiquetas de asignación de costos de Amazon SQS.

Antes de ejecutar el código de ejemplo, asegúrese de haber establecido sus AWS credenciales. Para obtener más información, consulte Configurar AWS las credenciales y la región para el desarrollo en la Guía para AWS SDK for Java 2.x desarrolladores.

Enumeración de etiquetas

Para obtener una lista de las etiquetas de una cola, utilice el método `ListQueueTags`.

```
// Create an SqsClient for the specified region.
SqsClient sqsClient = SqsClient.builder().region(Region.US_WEST_1).build();

// Get the queue URL.
String queueName = "MyStandardQ1";
GetQueueUrlResponse getQueueUrlResponse =

 sqsClient.getQueueUrl(GetQueueUrlRequest.builder().queueName(queueName).build());
String queueUrl = getQueueUrlResponse.queueUrl();

// Create the ListQueueTagsRequest.
final ListQueueTagsRequest listQueueTagsRequest =

 ListQueueTagsRequest.builder().queueUrl(queueUrl).build();

// Retrieve the list of queue tags and print them.
final ListQueueTagsResponse listQueueTagsResponse =
                                  sqsClient.listQueueTags(listQueueTagsRequest);
System.out.println(String.format("ListQueueTags: \tTags for queue %s are %s.\n",
                queueName, listQueueTagsResponse.tags() ));
```

Adición o actualización de etiquetas

Para agregar o actualizar valores de etiqueta para una cola, utilice el método `TagQueue`.

```
 // Create an SqsClient for the specified Region.
```

```
SqsClient sqsClient = SqsClient.builder().region(Region.US_WEST_1).build();

// Get the queue URL.
String queueName = "MyStandardQ1";
GetQueueUrlResponse getQueueUrlResponse =

 sqsClient.getQueueUrl(GetQueueUrlRequest.builder().queueName(queueName).build());
String queueUrl = getQueueUrlResponse.queueUrl();

// Build a hashmap of the tags.
final HashMap<String, String> addedTags = new HashMap<>();
        addedTags.put("Team", "Development");
        addedTags.put("Priority", "Beta");
        addedTags.put("Accounting ID", "456def");

//Create the TagQueueRequest and add them to the queue.
final TagQueueRequest tagQueueRequest = TagQueueRequest.builder()
        .queueUrl(queueUrl)
        .tags(addedTags)
        .build();
sqsClient.tagQueue(tagQueueRequest);
```

Eliminación de etiquetas

Para eliminar una o varias etiquetas de la cola, utilice el método `UntagQueue`. En el siguiente ejemplo se elimina la etiqueta `Accounting ID`.

```

// Create the UntagQueueRequest.
final UntagQueueRequest untagQueueRequest = UntagQueueRequest.builder()
        .queueUrl(queueUrl)
        .tagKeys("Accounting ID")
        .build();

// Remove the tag from this queue.
sqsClient.untagQueue(untagQueueRequest);
```

Envío de atributos de mensajes a una cola de Amazon SQS

Puede incluir metadatos estructurados (como marcas temporales, datos geoespaciales, firmas e identificadores) con los mensajes mediante atributos de mensaje. Para obtener más información, consulte Atributos de mensajes de Amazon SQS.

Antes de ejecutar el código de ejemplo, asegúrese de haber establecido sus AWS credenciales. Para obtener más información, consulte Configurar AWS las credenciales y la región para el desarrollo en la Guía para AWS SDK for Java 2.x desarrolladores.

Definición de atributos

Para definir un atributo para un mensaje, agregue el siguiente código, que utiliza el tipo de datos `MessageAttributeValue`. Para obtener más información, consulte Componentes de atributos de mensajes y Tipos de datos de atributos de mensajes.

Calcula AWS SDK for Java automáticamente las sumas de comprobación del cuerpo y los atributos del mensaje y las compara con los datos que devuelve Amazon SQS. Para obtener más información, consulte la Guía para desarrolladores de AWS SDK for Java 2.x y Cálculo del resumen del mensaje MD5 para atributos de mensajes para otros lenguajes de programación.

String

En este ejemplo se define un atributo `String` denominado `Name` con el valor `Jane`.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("Name", new MessageAttributeValue()
.withDataType("String")
.withStringValue("Jane"));
```

Number

En este ejemplo se define un atributo `Number` denominado `AccurateWeight` con el valor `230.00000000000000001`.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("AccurateWeight", new MessageAttributeValue()
.withDataType("Number")
.withStringValue("230.00000000000000001"));
```

Binary

En este ejemplo se define un atributo `Binary` denominado `ByteArray` con el valor de una matriz de 10 bytes sin inicializar.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("ByteArray", new MessageAttributeValue()
.withDataType("Binary")
.withBinaryValue(ByteBuffer.wrap(new byte[10])));
```

String (custom)

En este ejemplo se define el atributo personalizado `String.EmployeeId` denominado `EmployeeId` con el valor `ABC123456`.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("EmployeeId", new MessageAttributeValue()
.withDataType("String.EmployeeId")
.withStringValue("ABC123456"));
```

Number (custom)

En este ejemplo se define el atributo personalizado `Number.AccountId` denominado `AccountId` con el valor `000123456`.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("AccountId", new MessageAttributeValue()
.withDataType("Number.AccountId")
.withStringValue("000123456"));
```

> **Note**
>
> Dado que el tipo de datos base es `Number`, el método `ReceiveMessage` devuelve `123456`.

Binary (custom)

En este ejemplo se define el atributo personalizado `Binary.JPEG` denominado `ApplicationIcon` con el valor de una matriz de 10 bytes sin inicializar.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("ApplicationIcon", new MessageAttributeValue()
.withDataType("Binary.JPEG")
.withBinaryValue(ByteBuffer.wrap(new byte[10])));
```

Envío de un mensaje con atributos

En este ejemplo se agregan los atributos a `SendMessageRequest` antes de enviar el mensaje.

```
// Send a message with an attribute.
final SendMessageRequest sendMessageRequest = new SendMessageRequest();
sendMessageRequest.withMessageBody("This is my message text.");
sendMessageRequest.withQueueUrl(myQueueUrl);
sendMessageRequest.withMessageAttributes(messageAttributes);
sqs.sendMessage(sendMessageRequest);
```

> **Important**
>
> Si envía un mensaje a una cola FIFO (primero en entrar, primero en salir), asegúrese de que el método `sendMessage` se ejecuta después de proporcionar el ID del grupo de mensajes. Si utiliza el método `SendMessageBatch` en lugar de `SendMessage`, debe especificar los atributos de los mensajes del lote.

Uso de las API de Amazon SQS

En esta sección se proporciona información sobre cómo crear puntos de conexión de Amazon SQS, realizar solicitudes de API de consulta mediante los métodos GET y POST y utilizar acciones de API por lotes. Para obtener información detallada sobre las acciones de Amazon SQS (como parámetros, errores, ejemplos y tipos de datos), consulte la Referencia de la API de Amazon Simple Queue Service.

Si desea obtener acceso a Amazon SQS mediante diferentes lenguajes de programación, también puede utilizar los AWS SDK, que contienen la siguiente funcionalidad automática:

- Firmar criptográficamente sus solicitudes de servicio
- Reintentar solicitudes
- Tratar las respuestas a errores

Para obtener información sobre la herramienta de línea de comandos, consulte las secciones de Amazon SQS en la Referencia de comandos de AWS CLI y la Referencia de cmdlets de AWS Tools for PowerShell .

API de Amazon SQS con protocolo JSON AWS

Amazon SQS utiliza el protocolo AWS JSON como mecanismo de transporte para todas las API de Amazon SQS de las versiones del AWS SDK especificadas. AWS El protocolo JSON proporciona un mayor rendimiento, menor latencia y una comunicación más rápida. application-to-application AWS El protocolo JSON es más eficiente en la serialización o deserialización de solicitudes y respuestas en comparación con el protocolo de consulta. AWS Si aún prefiere utilizar el protocolo de AWS consulta con las API de SQS, consulte ¿Qué lenguajes son compatibles con el protocolo AWS JSON que se utiliza en las API de Amazon SQS? las versiones del AWS SDK que admiten el protocolo de consulta Amazon AWS SQS.

Amazon SQS utiliza el protocolo AWS JSON para comunicarse entre los clientes AWS del SDK (por ejemplo, Java, Python, Golang JavaScript) y el servidor Amazon SQS. Una solicitud HTTP de una operación de la API de Amazon SQS acepta la entrada con formato JSON. La operación de Amazon SQS se ejecuta y la respuesta de la ejecución se comparte de nuevo con el cliente del SDK en formato JSON. En comparación con las AWS consultas, AWS JSON es más simple, rápido y eficiente para transportar datos entre el cliente y el servidor.

- AWS El protocolo JSON actúa como mediador entre el cliente y el servidor de Amazon SQS.
- El servidor no entiende el lenguaje de programación en el que se crea la operación de Amazon SQS, pero entiende el protocolo AWS JSON.
- El protocolo AWS JSON utiliza la serialización (convierte el objeto a formato JSON) y la deserialización (convierte el formato JSON en objeto) entre el cliente y el servidor de Amazon SQS.

Para obtener más información sobre el protocolo AWS JSON con Amazon SQS, consulte. Preguntas frecuentes sobre el protocolo AWS JSON de Amazon SQS

AWS El protocolo JSON está disponible en la versión del AWS SDK especificada. Para consultar la versión y las fechas de lanzamiento del SDK en las distintas variantes de lenguaje, consulte Matriz de compatibilidad para versiones de SDK y herramientas de AWS en la Guía de referencia de los SDK y las herramientas de AWS .

Temas

- Realizar solicitudes de API de consulta mediante el protocolo AWS JSON en Amazon SQS
- Realizar solicitudes de API de AWS consultas mediante el protocolo de consultas en Amazon SQS
- Autenticación de solicitudes para Amazon SQS
- Acciones por lotes de Amazon SQS

Realizar solicitudes de API de consulta mediante el protocolo AWS JSON en Amazon SQS

En esta sección, aprenderá a crear un punto de conexión de Amazon SQS, a crear solicitudes POST y a interpretar las respuestas.

> **Note**
>
> AWS El protocolo JSON es compatible con la mayoría de las variantes de idioma. Para ver una lista completa de las variantes de lenguaje admitidas, consulte ¿Qué lenguajes son compatibles con el protocolo AWS JSON que se utiliza en las API de Amazon SQS?.

Temas

- Construcción de un punto de enlace
- Realizar una solicitud POST
- Interpretación de las respuestas de la API JSON de Amazon SQS
- Preguntas frecuentes sobre el protocolo AWS JSON de Amazon SQS

Construcción de un punto de enlace

Para trabajar con colas de Amazon SQS, debe crear un punto de conexión. Para obtener información sobre los puntos de conexión de Amazon SQS, consulte las páginas siguientes en Referencia general de Amazon Web Services:

- Puntos de conexión regionales
- Puntos de conexión y cuotas de Amazon Simple Queue Service

Cada punto de conexión de Amazon SQS es independiente. Por ejemplo, si dos colas tienen un nombre MyQueuey una tiene el punto final `sqs.us-east-2.amazonaws.com` mientras que la otra tiene el punto final`sqs.eu-west-2.amazonaws.com`, las dos colas no comparten ningún dato entre sí.

A continuación, se muestra un ejemplo de un punto de conexión que realiza una solicitud para crear una cola.

```
POST / HTTP/1.1
Host: sqs.us-west-2.amazonaws.com
X-Amz-Target: AmazonSQS.CreateQueue
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueName":"MyQueue",
    "Attributes": {
        "VisibilityTimeout": "40"
    },
    "tags": {
        "QueueType": "Production"
    }
```

```
}
```

> **Note**
>
> Los nombres y las URL de las colas distinguen entre mayúsculas y minúsculas.
> La estructura de *AUTHPARAMS* depende de la firma de la solicitud de API. Para obtener más información, consulte Firmar solicitudes de AWS API en la Referencia general de Amazon Web Services.

Realizar una solicitud POST

Las solicitudes POST de Amazon SQS envían parámetros de consulta como un formulario en el cuerpo de una solicitud HTTP.

A continuación, se muestra un ejemplo de un encabezado HTTP con `X-Amz-Target` establecido a `AmazonSQS.<operationName>` y de un encabezado HTTP con `Content-Type` establecido a `application/x-amz-json-1.0`.

```
POST / HTTP/1.1
Host: sqs.<region>.<domain>
X-Amz-Target: AmazonSQS.SendMessage
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.<region>.<domain>/<awsAccountId>/<queueName>/",
    "MessageBody": "This is a test message",
}
```

Esta solicitud HTTP POST envía un mensaje a una cola de Amazon SQS.

> **Note**
>
> Ambos encabezados HTTP `X-Amz-Target` y `Content-Type` son obligatorios.
> El cliente HTTP puede añadir otros elementos a la solicitud HTTP, según la versión de HTTP del cliente.

Interpretación de las respuestas de la API JSON de Amazon SQS

En respuesta a una solicitud de acción, Amazon SQS devuelve una estructura de datos XML que contiene los resultados de la solicitud. Para obtener más información, consulte las acciones individuales en la Referencia de la API de Amazon Simple Queue Service y Preguntas frecuentes sobre el protocolo AWS JSON de Amazon SQS.

Temas

Estructura de una respuesta JSON correcta

Si la solicitud se realiza correctamente, el elemento de respuesta principal es `x-amzn-RequestId`, que contiene el identificador único universal (UUID) de la solicitud, así como otros campos de respuesta añadidos. Por ejemplo, la siguiente respuesta de `CreateQueue` contiene el campo `QueueUrl` que, a su vez, contiene la URL de la cola creada.

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "QueueUrl":"https://sqs.us-east-1.amazonaws.com/111122223333/MyQueue"
}
```

Estructura de una respuesta de error JSON

Si una solicitud no se realiza correctamente, Amazon SQS devuelve la respuesta principal, incluidos el encabezado HTTP y el cuerpo.

En el encabezado HTTP, `x-amzn-RequestId` contiene el UUID de la solicitud. `x-amzn-query-error` contiene dos informaciones: el tipo de error y si se trata de un error del productor o del consumidor.

En el cuerpo de la respuesta, `"__type"` indica otros detalles del error y `Message` señala la condición del error en un formato legible.

El siguiente es un ejemplo de respuesta de error en formato JSON:

```
HTTP/1.1 400 Bad Request
x-amzn-RequestId: 66916324-67ca-54bb-a410-3f567a7a0571
x-amzn-query-error: AWS.SimpleQueueService.NonExistentQueue;Sender
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "__type": "com.amazonaws.sqs#QueueDoesNotExist",
    "message": "The specified queue does not exist."
}
```

Preguntas frecuentes sobre el protocolo AWS JSON de Amazon SQS

Preguntas frecuentes sobre el uso del protocolo AWS JSON con Amazon SQS.

¿Qué es el protocolo AWS JSON y en qué se diferencia de las solicitudes y respuestas de la API Amazon SQS existentes?

JSON es uno de los métodos de conexión más utilizados y aceptados para la comunicación entre sistemas heterogéneos. Amazon SQS utiliza JSON como medio de comunicación entre un cliente AWS del SDK (por ejemplo, Java, Python, Golang JavaScript) y el servidor Amazon SQS. Una solicitud HTTP de una operación de la API de Amazon SQS acepta entradas en formato JSON. La operación de Amazon SQS se ejecuta y la respuesta de la ejecución se comparte de nuevo con el cliente del SDK en formato JSON. En comparación con las consultas de AWS , JSON es más eficiente a la hora de transportar datos entre el cliente y el servidor.

- El protocolo Amazon SQS AWS JSON actúa como mediador entre el cliente y el servidor de Amazon SQS.
- El servidor no entiende el lenguaje de programación en el que se crea la operación de Amazon SQS, pero entiende el protocolo AWS JSON.
- El protocolo Amazon SQS AWS JSON utiliza la serialización (convierte el objeto a formato JSON) y la deserialización (convierte el formato JSON en objeto) entre el cliente y el servidor de Amazon SQS.

¿Cómo puedo empezar a utilizar los protocolos AWS JSON para Amazon SQS?

Para empezar a utilizar la última versión del AWS SDK y lograr una mensajería más rápida para Amazon SQS, actualice el AWS SDK a la versión especificada o a cualquier versión posterior. Para obtener más información sobre los clientes del SDK, consulte la columna Guía de la tabla siguiente.

La siguiente es una lista de las versiones del SDK en todas las variantes de idioma del protocolo AWS JSON para su uso con las API de Amazon SQS:

Idioma	Repositorio de clientes del SDK	Versión de cliente del SDK requerida	Guía
C++	aws/aws-sdk-cpp	1.11.98	AWS SDK para C++
Golang 1.x	aws/aws-sdk-go	v1.47.7	AWS SDK para Go
Golang 2.x	aws/aws-sdk-go-v2	v1.28.0	AWS SDK para Go V2
Java 1.x	aws/aws-sdk-java	1.12.585	AWS SDK para Java
Java 2.x	aws/aws-sdk-java-v2	2.21.19	AWS SDK para Java
JavaScript v2.x	aws/aws-sdk-js	v2.1492.0	JavaScript en AWS
JavaScript v3.x	aws/aws-sdk-js-v3	v3.447.0	JavaScript en AWS
.NET	aws/aws-sdk-net	3.7.681.0	AWS SDK para .NET
PHP	aws/aws-sdk-php	3.285.2	AWS SDK para PHP
Python-boto3	boto/boto3	1.28,82	AWS SDK para Python (Boto3)

Idioma	Repositorio de clientes del SDK	Versión de cliente del SDK requerida	Guía
Python-botocore	boto/botocore	1,31,82	AWS SDK para Python (Boto3)
awscli	CLI de AWS	1.29.82	Interfaz de la línea de comandos de AWS
Ruby	aws/aws-sdk-ruby	1.67,0	AWS SDK para Ruby

¿Cuáles son los riesgos de habilitar el protocolo JSON para mis cargas de trabajo de Amazon SQS?

Si utiliza una implementación personalizada del AWS SDK o una combinación de clientes personalizados y un AWS SDK para interactuar con Amazon SQS que genera respuestas basadas en AWS consultas (también conocidas como basadas en XML), es posible que no sea compatible con el protocolo JSON. AWS Si tiene algún problema, póngase en contacto con AWS Support.

¿Qué sucede si ya tengo la última versión del AWS SDK, pero mi solución de código abierto no es compatible con JSON?

Debe cambiar la versión del SDK por la anterior a la que esté utilizando. Consulte ¿Cómo puedo empezar a utilizar los protocolos AWS JSON para Amazon SQS? para obtener más información. AWS Las versiones del SDK que aparecen en ¿Cómo puedo empezar a utilizar los protocolos AWS JSON para Amazon SQS? utilizan el protocolo JSON wire para las API de Amazon SQS. Si cambia el AWS SDK a la versión anterior, las API de Amazon SQS utilizarán la AWS consulta.

¿Qué lenguajes son compatibles con el protocolo AWS JSON que se utiliza en las API de Amazon SQS?

Amazon SQS admite todas las variantes de idioma en las que los AWS SDK están disponibles de forma general (GA). Actualmente, no se admite Kotlin, Rust ni Swift. Para obtener más información sobre otras variantes de lenguaje, consulte Herramientas para crear en AWS.

Qué regiones se admiten en el protocolo AWS JSON que se utiliza en las API de Amazon SQS

Amazon SQS admite el protocolo AWS JSON en todas AWS las regiones en las que Amazon SQS está disponible.

¿Qué mejoras de latencia puedo esperar al actualizar a las versiones de AWS SDK especificadas para Amazon SQS mediante el protocolo AWS JSON?

AWS El protocolo JSON es más eficiente en la serialización y deserialización de solicitudes y respuestas en comparación con el protocolo de consultas. AWS Según las pruebas de AWS rendimiento para una carga útil de mensajes de 5 KB, el protocolo JSON para Amazon SQS end-to-end reduce la latencia del procesamiento de mensajes hasta un 23% y reduce el uso de CPU y memoria del lado del cliente de la aplicación.

¿Quedará AWS obsoleto el protocolo de consultas?

AWS el protocolo de consulta seguirá siendo compatible. Puede seguir utilizando el protocolo de AWS consultas siempre que la versión AWS del SDK esté configurada en una versión anterior distinta de la que se indica en Cómo empezar con los protocolos AWS JSON para Amazon SQS.

¿Dónde puedo encontrar más información sobre el protocolo AWS JSON?

Puede encontrar más información sobre el protocolo JSON en Protocolo AWS JSON 1.0 en la documentación de Smithy. Para obtener más información sobre las solicitudes de la API de Amazon SQS mediante el protocolo AWS JSON, consulte Realizar solicitudes de API de consulta mediante el protocolo AWS JSON en Amazon SQS.

Realizar solicitudes de API de AWS consultas mediante el protocolo de consultas en Amazon SQS

En esta sección, aprenderá a crear un punto de conexión de Amazon SQS, a crear solicitudes GET y POST y a interpretar las respuestas.

Temas

- Construcción de un punto de enlace
- Realizar una solicitud GET
- Realizar una solicitud POST

- Interpretación de las respuestas de la API XML de Amazon SQS

Construcción de un punto de enlace

Para trabajar con colas de Amazon SQS, debe crear un punto de conexión. Para obtener información sobre los puntos de conexión de Amazon SQS, consulte las páginas siguientes en Referencia general de Amazon Web Services:

- Puntos de conexión regionales
- Puntos de conexión y cuotas de Amazon Simple Queue Service

Cada punto de conexión de Amazon SQS es independiente. Por ejemplo, si dos colas tienen un nombre MyQueuey una tiene el punto final `sqs.us-east-2.amazonaws.com` mientras que la otra tiene el punto final`sqs.eu-west-2.amazonaws.com`, las dos colas no comparten ningún dato entre sí.

A continuación se muestra un ejemplo de un punto de enlace que realiza una solicitud para crear una cola.

```
https://sqs.eu-west-2.amazonaws.com/
?Action=CreateQueue
&DefaultVisibilityTimeout=40
&QueueName=MyQueue
&Version=2012-11-05
&AUTHPARAMS
```

> **Note**
>
> Los nombres y las URL de las colas distinguen entre mayúsculas y minúsculas.
> La estructura de *AUTHPARAMS* depende de la firma de la solicitud de API. Para obtener más información, consulte Firmar solicitudes de AWS API en la Referencia general de Amazon Web Services.

Realizar una solicitud GET

Una solicitud GET de Amazon SQS se estructura como una URL que consta de los siguientes elementos:

- Punto de conexión: el recurso sobre el que actúa la solicitud (el nombre y la URL de la cola); por ejemplo, `https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue`
- Acción: la acción que se desea realizar en el punto de conexión. Se utiliza un signo de interrogación (?) para separar el punto de enlace de la acción; por ejemplo, `?Action=SendMessage&MessageBody=Your%20Message%20Text`
- Parámetros: los parámetros de la solicitud. Cada parámetro está separado del siguiente por el signo &; por ejemplo, `&Version=2012-11-05&AUTHPARAMS`

A continuación se muestra un ejemplo de una solicitud GET que envía un mensaje a una cola de Amazon SQS.

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue
?Action=SendMessage&MessageBody=Your%20message%20text
&Version=2012-11-05
&AUTHPARAMS
```

> **Note**
>
> Los nombres y las URL de las colas distinguen entre mayúsculas y minúsculas.
> Como las solicitudes GET son direcciones URL, debe codificar como URL los valores de todos los parámetros. Debido a que en las URL no se permiten los espacios en blanco, que cada espacio se codifica en la URL como `%20`. El resto del ejemplo no se ha codificado como URL para facilitar su lectura.

Realizar una solicitud POST

Las solicitudes POST de Amazon SQS envían parámetros de consulta como un formulario en el cuerpo de una solicitud HTTP.

A continuación, se muestra un ejemplo de encabezado HTTP con `Content-Type` establecido a `application/x-www-form-urlencoded`.

```
POST /123456789012/MyQueue HTTP/1.1
Host: sqs.us-east-2.amazonaws.com
Content-Type: application/x-www-form-urlencoded
```

Al encabezado le sigue una solicitud GET de `form-urlencoded` que envía un mensaje a una cola de Amazon SQS. Cada parámetro está separado del siguiente por un signo ampersand (&).

```
Action=SendMessage
&MessageBody=Your+Message+Text
&Expires=2020-10-15T12%3A00%3A00Z
&Version=2012-11-05
&AUTHPARAMS
```

> **Note**
>
> Solo es obligatorio el encabezado de HTTP `Content-Type`. El parámetro *`AUTHPARAMS`* es el mismo que para la solicitud GET.
> El cliente HTTP puede añadir otros elementos a la solicitud HTTP, según la versión de HTTP del cliente.

Interpretación de las respuestas de la API XML de Amazon SQS

Como respuesta a una solicitud de acción, Amazon SQS devuelve una estructura de datos XML que contiene los resultados de la solicitud. Para obtener más información, consulte las acciones individuales en la Referencia de la API de Amazon Simple Queue Service.

Temas

- Estructura de una respuesta XML correcta
- Estructura de una respuesta de error XML

Estructura de una respuesta XML correcta

Si la solicitud se realiza correctamente, el elemento principal de la respuesta tiene el mismo nombre que la acción, pero se le añade `Response` (por ejemplo, *`ActionName`*`Response`).

Este elemento contiene los siguientes elementos secundarios:

- ***`ActionName`*`Result`**: contiene un elemento específico de la acción. Por ejemplo, el elemento `CreateQueueResult` contiene el elemento `QueueUrl` que, a su vez, contiene la URL de la cola que se ha creado.

- **`ResponseMetadata`**: contiene el `RequestId` que, a su vez, contiene el identificador único universal (UUID) de la solicitud.

A continuación se muestra un ejemplo de una respuesta correcta en formato XML:

```
<CreateQueueResponse
   xmlns=https://sqs.us-east-2.amazonaws.com/doc/2012-11-05/
   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
   xsi:type=CreateQueueResponse>
   <CreateQueueResult>
      <QueueUrl>https://sqs.us-east-2.amazonaws.com/770098461991/queue2</QueueUrl>
   </CreateQueueResult>
   <ResponseMetadata>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
   </ResponseMetadata>
</CreateQueueResponse>
```

Estructura de una respuesta de error XML

Si una solicitud no se realiza correctamente, Amazon SQS siempre devuelve el elemento principal de la respuesta `ErrorResponse`. Este elemento contiene un elemento `Error` y un elemento `RequestId`.

El elemento `Error` contiene los siguientes elementos secundarios:

- **`Type`**: especifica si se trata de un error del productor o del consumidor.
- **`Code`**: especifica el tipo de error.
- **`Message`**: especifica la condición de error en formato legible.
- **`Detail`**: (opcional) especifica información adicional sobre el error.

El elemento `RequestId` contiene el UUID de la solicitud.

A continuación se muestra un ejemplo de una respuesta de error en formato XML:

```
<ErrorResponse>
   <Error>
      <Type>Sender</Type>
      <Code>InvalidParameterValue</Code>
      <Message>
         Value (quename_nonalpha) for parameter QueueName is invalid.
```

```
         Must be an alphanumeric String of 1 to 80 in length.
      </Message>
   </Error>
   <RequestId>42d59b56-7407-4c4a-be0f-4c88daeea257</RequestId>
</ErrorResponse>
```

Autenticación de solicitudes para Amazon SQS

La autenticación es un proceso para identificar y verificar a la parte que envía una solicitud. Durante la primera fase de la autenticación, AWS verifica la identidad del productor y si está registrado para utilizar AWS (para obtener más información, consulte Paso 1: Crear un usuario Cuenta de AWS de IAM). A continuación AWS , sigue el siguiente procedimiento:

1. El productor (remitente) obtiene la credencial necesaria.
2. El productor envía una solicitud y la credencial al consumidor (receptor).
3. El consumidor utiliza la credencial para verificar si el productor envió la solicitud.
4. Se produce una de las circunstancias siguientes:
   - Si la autenticación se realiza correctamente, el consumidor procesa la solicitud.
   - Si se produce un error de autenticación, el consumidor rechaza la solicitud y devuelve un error.

Temas

Proceso básico de autenticación con HMAC-SHA

Cuando se obtiene acceso a Amazon SQS con la API de consultas, se deben proporcionar los siguientes elementos para autenticar la solicitud:

- El identificador de clave de AWS acceso que lo identifica Cuenta de AWS y que se AWS utiliza para buscar su clave de acceso secreta.
- La firma de la solicitud HMAC-SHA, que se calcula utilizando la clave de acceso secreta (un secreto compartido conocido únicamente por usted y por AWS; para obtener más información,

consulte RFC2104). El AWS SDK se encarga del proceso de firma; sin embargo, si se realiza una solicitud de consulta a través de HTTP o HTTPS, se deberá incluir una firma en cada solicitud de consulta.

1. Genere una clave firma Signature Version 4. Para obtener más información, consulte Generación de la clave de firma con Java.

   > **Note**
   >
   > Amazon SQS admite Signature Version 4, que proporciona una seguridad basada en SHA256 y un rendimiento mejorados con respecto a las versiones anteriores. Cuando cree aplicaciones nuevas que usen Amazon SQS, utilice Signature Version 4.

2. codifique en formato base64 la firma de la solicitud. El siguiente ejemplo se de código Java lo hace:

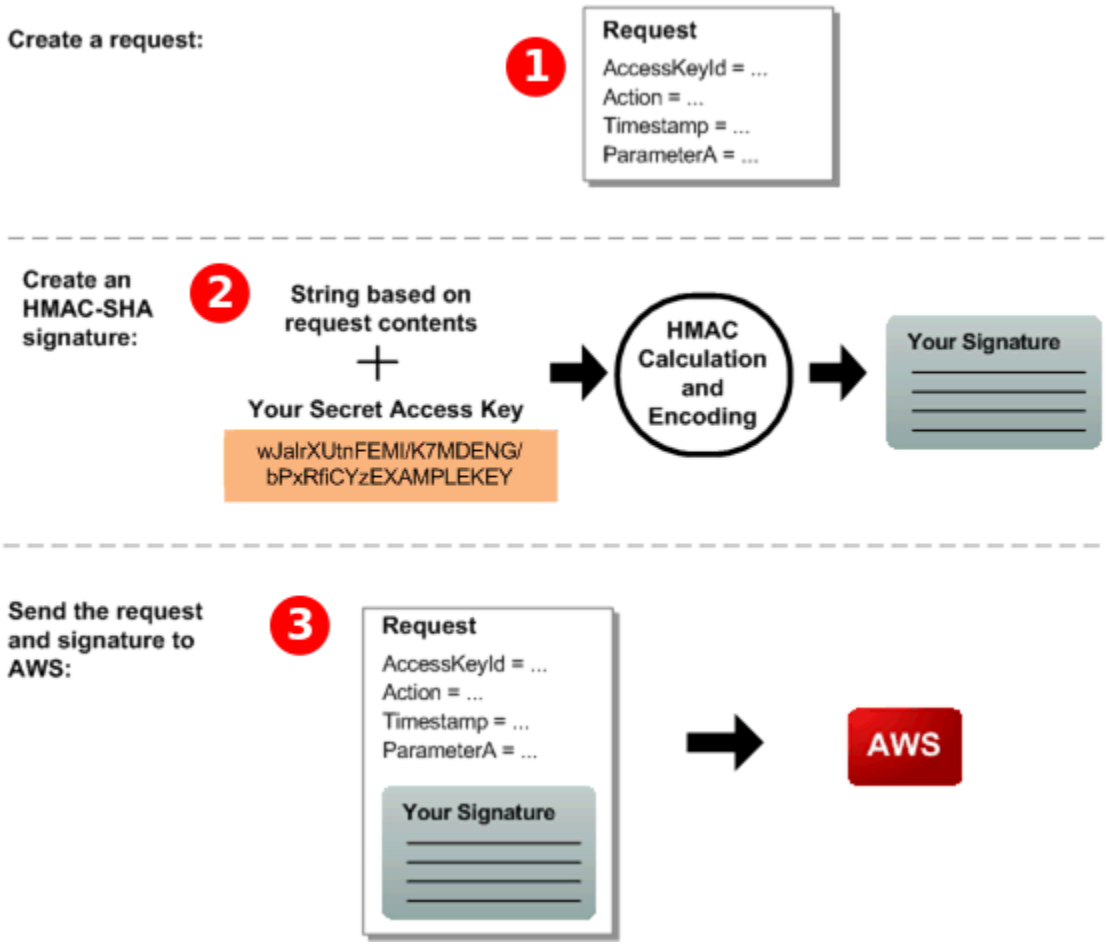
```
package amazon.webservices.common;

// Define common routines for encoding data in AWS requests.
public class Encoding {

    /* Perform base64 encoding of input bytes.
     * rawData is the array of bytes to be encoded.
     * return is the base64-encoded string representation of rawData.
     */
    public static String EncodeBase64(byte[] rawData) {
        return Base64.encodeBytes(rawData);
    }
}
```

- La marca temporal (o vencimiento) de la solicitud. La marca temporal que utilice en la solicitud debe ser un objeto `dateTime`, con la fecha completa, incluidas las horas, los minutos y los segundos. Por ejemplo: `2007-01-31T23:59:59Z` Aunque no es necesario, le recomendamos que proporcione el objeto utilizando la zona horaria de la hora universal coordinada (hora del meridiano de Greenwich).

> **Note**
>
> Asegúrese de que la hora del servidor esté ajustada correctamente. Si especificas una marca de tiempo (en lugar de una fecha de caducidad), la solicitud caduca automáticamente 15 minutos después de la hora especificada (AWS no procesa las solicitudes con marcas de tiempo más de 15 minutos antes de la hora actual en los servidores). AWS
> Si utiliza .NET, no debe enviar marcas temporales demasiado específicas (debido a las diferentes interpretaciones con respecto a cómo se debe descartar la precisión adicional del tiempo). En este caso, debe crear manualmente objetos `dateTime` con una precisión de no más de un milisegundo.

Parte 1: la solicitud del usuario

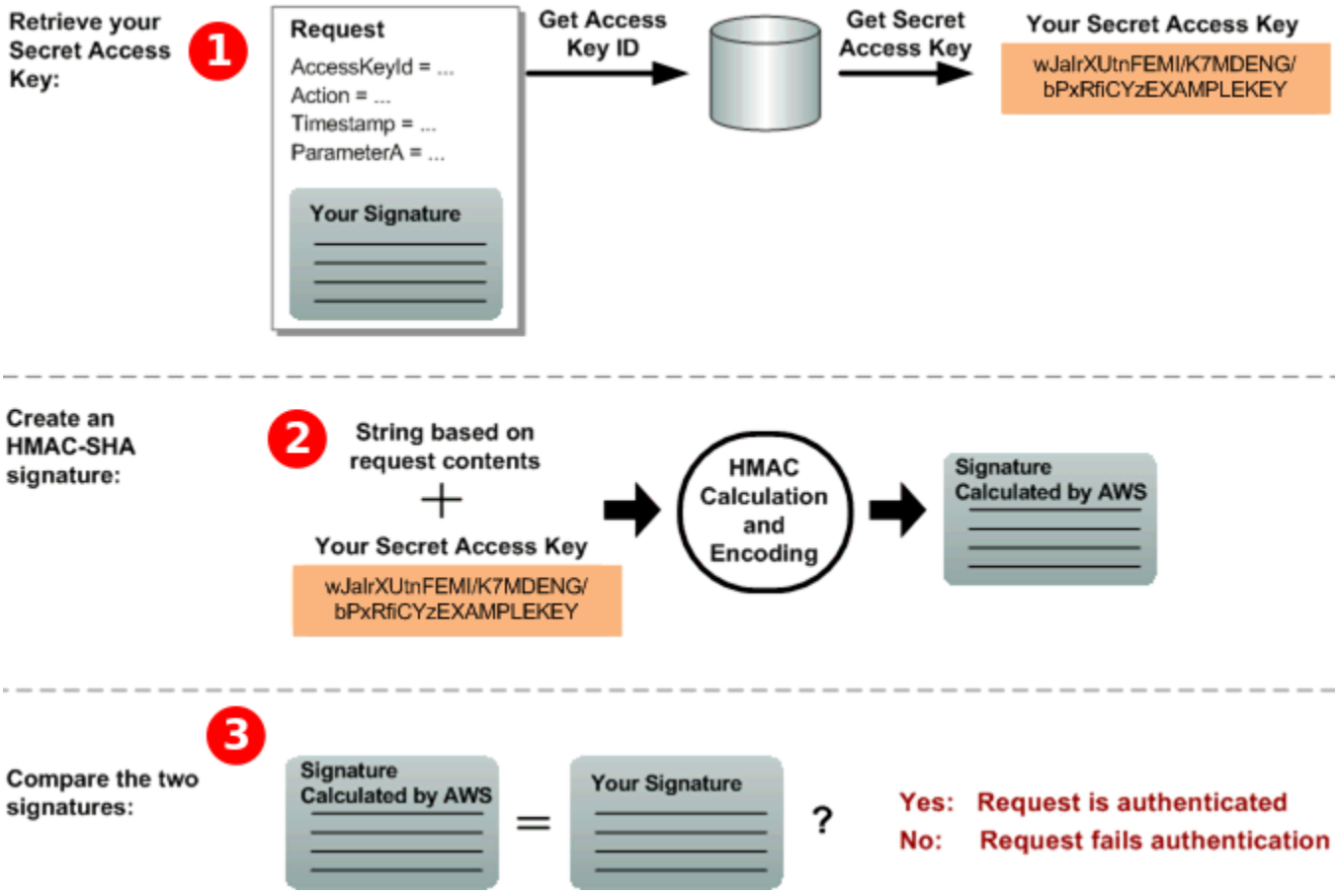
El siguiente es el proceso que debe seguir para autenticar las AWS solicitudes mediante una firma de solicitud del HMAC-SHA.



1. Cree una solicitud para AWS.
2. Calcule la firma de código de autenticación de mensajes mediante algoritmos hash con clave (HMAC-SHA) utilizando su clave de acceso secreta.
3. Incluye la firma y tu ID de clave de acceso en la solicitud y, a continuación, envía la solicitud a AWS.

Parte 2: La respuesta de AWS

AWS comienza el siguiente proceso en respuesta.



1. AWS utiliza el identificador de clave de acceso para buscar su clave de acceso secreta.
2. AWS genera una firma a partir de los datos de la solicitud y la clave de acceso secreta, utilizando el mismo algoritmo que utilizó para calcular la firma que envió en la solicitud.
3. Se produce una de las circunstancias siguientes:
   - Si la firma que se AWS genera coincide con la que se envió en la solicitud, se considerará AWS que la solicitud es auténtica.
   - Si la comparación falla, la solicitud se descarta y AWS devuelve un error.

Acciones por lotes de Amazon SQS

Para reducir los costos o manipular hasta 10 mensajes con una única acción, puede utilizar las siguientes acciones:

- `SendMessageBatch`
- `DeleteMessageBatch`

- `ChangeMessageVisibilityBatch`

Puede aprovechar la funcionalidad por lotes mediante la API de consultas o un AWS SDK que admita las acciones por lotes de Amazon SQS.

> **Note**
>
> El tamaño total de todos los mensajes que envíes en una sola `SendMessageBatch` llamada no puede superar los 262.144 bytes (256 KiB).
> No se pueden establecer permisos para `SendMessageBatch`, `DeleteMessageBatch` ni `ChangeMessageVisibilityBatch` de forma explícita. Al establecer permisos para `SendMessage`, `DeleteMessage` o `ChangeMessageVisibility`, también se establecen permisos para las versiones por lotes correspondientes de esas acciones.
> La consola de Amazon SQS no es compatible con las acciones de procesamiento por lotes.

Temas

- Habilitar el almacenamiento en búfer del lado del cliente y el procesamiento por lotes de solicitudes con Amazon SQS
- Aumentar el rendimiento mediante el escalado horizontal y el procesamiento por lotes de acciones con Amazon SQS

Habilitar el almacenamiento en búfer del lado del cliente y el procesamiento por lotes de solicitudes con Amazon SQS

AWS SDK for Java incluye `AmazonSQSBufferedAsyncClient` que tiene acceso a Amazon SQS. Este cliente permite agrupar fácilmente en lotes las solicitudes mediante el almacenamiento en búfer en el cliente, en el que las llamadas realizadas desde el cliente primero se almacenan en búfer y después se envían como una solicitud por lotes a Amazon SQS.

El almacenamiento en búfer en el cliente permite almacenar en búfer hasta diez solicitudes y enviarlas como una solicitud por lotes, lo que disminuye el costo de uso de Amazon SQS y el número de solicitudes enviadas. `AmazonSQSBufferedAsyncClient` almacena en búfer tanto las llamadas sincrónicas como las asincrónicas. Las solicitudes por lotes y la compatibilidad con los sondeos largos también pueden contribuir a mejorar el rendimiento. Para obtener más información, consulte

Aumentar el rendimiento mediante el escalado horizontal y el procesamiento por lotes de acciones con Amazon SQS.

Dado que `AmazonSQSBufferedAsyncClient` implementa la misma interfaz que `AmazonSQSAsyncClient`, la migración de `AmazonSQSAsyncClient` a `AmazonSQSBufferedAsyncClient` solo suele requerir cambios mínimos en el código.

> **Note**
>
> El cliente asincrónico con búfer de Amazon SQS no admite actualmente las colas FIFO.

Temas

- Uso del cliente AmazonSQS BufferedAsync
- Configuración del cliente AmazonSQS BufferedAsync

Uso del cliente AmazonSQS BufferedAsync

Antes de comenzar, complete los pasos de Configuración de Amazon SQS.

> **Important**
>
> Actualmente, AWS SDK for Java 2.x no es compatible con. `AmazonSQSBufferedAsyncClient`

Puede crear un nuevo cliente `AmazonSQSBufferedAsyncClient` basado en `AmazonSQSAsyncClient`; por ejemplo:

```
// Create the basic Amazon SQS async client
final AmazonSQSAsync sqsAsync = new AmazonSQSAsyncClient();

// Create the buffered client
final AmazonSQSAsync bufferedSqs = new AmazonSQSBufferedAsyncClient(sqsAsync);
```

Después de crear el nuevo `AmazonSQSBufferedAsyncClient`, puede utilizarlo para enviar varias solicitudes a Amazon SQS (del mismo modo que con `AmazonSQSAsyncClient`), por ejemplo:

```
final CreateQueueRequest createRequest = new
 CreateQueueRequest().withQueueName("MyQueue");

final CreateQueueResult res = bufferedSqs.createQueue(createRequest);

final SendMessageRequest request = new SendMessageRequest();
final String body = "Your message text" + System.currentTimeMillis();
request.setMessageBody( body );
request.setQueueUrl(res.getQueueUrl());

final Future<SendMessageResult> sendResult = bufferedSqs.sendMessageAsync(request);

final ReceiveMessageRequest receiveRq = new ReceiveMessageRequest()
    .withMaxNumberOfMessages(1)
    .withQueueUrl(queueUrl);
final ReceiveMessageResult rx = bufferedSqs.receiveMessage(receiveRq);
```

Configuración del cliente AmazonSQS BufferedAsync

`AmazonSQSBufferedAsyncClient` está preconfigurado con ajustes válidos para la mayoría de los casos de uso. Se pueden configurar ajustes adicionales de `AmazonSQSBufferedAsyncClient`; por ejemplo:

1. Crear una instancia de la clase `QueueBufferConfig` con los parámetros de configuración necesarios.
2. Proporcionar la instancia al constructor `AmazonSQSBufferedAsyncClient`.

```
// Create the basic Amazon SQS async client
final AmazonSQSAsync sqsAsync = new AmazonSQSAsyncClient();

final QueueBufferConfig config = new QueueBufferConfig()
    .withMaxInflightReceiveBatches(5)
    .withMaxDoneReceiveBatches(15);

// Create the buffered client
final AmazonSQSAsync bufferedSqs = new AmazonSQSBufferedAsyncClient(sqsAsync, config);
```

QueueBufferConfig parámetros de configuración

Parámetro	Valor predeterminado	Descripción
`longPoll`	`true`	Cuando `longPoll` se establece en `true`, `AmazonSQSBufferedAsyncClient` intenta utilizar el sondeo largo a la hora de consumir mensajes.
`longPollWaitTimeoutSeconds`	20 s	El tiempo máximo, en segundos, que una llamada a `ReceiveMessage` se bloquea en el servidor a la espera de que aparezcan mensajes en la cola antes de devolver un resultado de recepción vacío. **Note** Este parámetro no tiene ningún efecto cuando el sondeo largo está deshabilitado.
`maxBatchOpenMs`	200ms	El tiempo máximo, en milisegundos, que una llamada saliente espera otras llamadas para procesar por lotes mensajes del mismo tipo. Cuanto mayor sea el valor, menos lotes son necesario

Parámetro	Valor predeterminado	Descripción
		s para realizar la misma cantidad de trabajo (no obstante, la primera llamada de un lote tiene que esperar más tiempo). Cuando se establece este parámetro en `0`, las solicitud es enviadas no esperan a otras solicitudes, lo que en la práctica deshabilita el procesamiento por lotes.
`maxBatchSize`	10 solicitudes por lote	El número máximo de mensajes que se procesan juntos por lotes en una sola solicitud. Cuanto mayor sea la configuración, menos lotes serán necesarios para llevar a cabo el mismo número de solicitudes. **Note** El valor máximo permitido para Amazon SQS es de diez solicitudes por lote.

Parámetro	Valor predeterminado	Descripción
`maxBatchSizeBytes`	256 KiB	El tamaño máximo de un lote de mensajes, en bytes, que el cliente intenta enviar a Amazon SQS. **Note** 256 KiB es el valor máximo permitido para Amazon SQS.

Parámetro	Valor predeterminado	Descripción
`maxDoneReceiveBatches`	10 lotes	El número máximo de lotes de recepción que `AmazonSQSBufferedAsyncClient` captura previamente y almacena en el lado del cliente. Cuanto mayor sea el valor, más solicitudes de recepción podrán satisfacerse sin tener que realizar una llamada a Amazon SQS (sin embargo, cuantos más mensajes se capturen previamente, más tiempo permanecerán en el búfer, lo que hará que caduque su propio tiempo de espera de visibilidad). **Note** `0`indica que la captura previa de todos los mensajes está deshabilitada y que los mensajes se consumen solo a pedido.

Parámetro	Valor predeterminado	Descripción
`maxInflightOutboundBatches`	5 lotes	El número máximo de lotes salientes activos que se pueden procesar al mismo tiempo. Cuanto mayor sea el valor, más rápido se podrán enviar los lotes salientes (sujeto a cuotas como la CPU o el ancho de banda) y más subprocesos podrá consumir `AmazonSQSBufferedAsyncClient` .

Parámetro	Valor predeterminado	Descripción
`maxInflightReceiveBatches`	10 lotes	El número máximo de lotes de recepción activos que se pueden procesar al mismo tiempo. Cuanto mayor sea el valor, más mensajes de podrán recibir (sujeto a cuotas como la CPU o el ancho de banda) y más subprocesos podrá consumir `AmazonSQSBufferedAsyncClient` . **Note** `0`indica que la captura previa de todos los mensajes está deshabilitada y que los mensajes se consumen solo a pedido.

Parámetro	Valor predeterminado	Descripción
`visibilityTimeoutSeconds`	-1	Cuando este parámetro se establece en un valor positivo distinto de cero, el tiempo de espera de visibilidad que se establece aquí anula el tiempo de espera de visibilidad definido en la cola desde la que se consumen los mensajes. **Note** `-1` indica que está seleccionada la configuración predeterminada de la cola. No se puede establecer el tiempo de espera de visibilidad en `0`.

Aumentar el rendimiento mediante el escalado horizontal y el procesamiento por lotes de acciones con Amazon SQS

Las colas de Amazon SQS pueden ofrecer un rendimiento muy elevado. Para obtener información sobre las cuotas de rendimiento, consulte Cuotas de mensajes de Amazon SQS.

Para conseguir un desempeño elevado, debe escalar horizontalmente los productores y los consumidores de mensajes (añadir más productores y consumidores).

Temas

- Escalado horizontal

- Agrupación en lotes de acciones
- Ejemplo funcional en Java de solicitudes de una sola operación y por lotes

Escalado horizontal

Dado que se accede a Amazon SQS a través de un protocolo HTTP de solicitud-respuesta, la latencia de solicitudes (el intervalo de tiempo que transcurre entre que se inicia una solicitud y se recibe una respuesta) limita el rendimiento que se puede alcanzar con un solo subproceso a través de una única conexión. Por ejemplo, si la latencia desde un cliente basado en Amazon EC2 hasta Amazon SQS en la misma región es de una media de 20 ms, el rendimiento máximo desde un único subproceso a través de una única conexión será de una media de 50 transacciones por segundo.

El escalado horizontal supone aumentar el número de productores de mensajes (que realizan solicitudes `SendMessage`) y consumidores de mensajes (que realizan solicitudes `ReceiveMessage` y `DeleteMessage`) para aumentar el desempeño general de la cola. Existen tres métodos para realizar el escalado horizontal:

- Aumentar el número de subprocesos por cliente
- Añadir más clientes
- Aumentar el número de subprocesos por cliente y añadir más clientes

Cuando se añaden más clientes, se consiguen ganancias prácticamente lineales en el desempeño de la cola. Por ejemplo, si se duplica el número de clientes, también se obtiene el doble de desempeño.

> **Note**
>
> A medida que realice el escalado horizontal, debe asegurarse de que el cliente de Amazon SQS tiene suficientes conexiones o subprocesos para admitir el número de productores de mensajes simultáneos y consumidores que envían solicitudes y reciben respuestas. Por ejemplo, de forma predeterminada, las instancias de la AWS SDK for Java `AmazonSQSClient` clase mantienen como máximo 50 conexiones a Amazon SQS. Para crear productores y consumidores adicionales simultáneos, debe ajustar el número máximo de subprocesos de productores y consumidores admisibles en un objeto `AmazonSQSClientBuilder`; por ejemplo:
>
> ```
> final AmazonSQS sqsClient = AmazonSQSClientBuilder.standard()
> ```

```
        .withClientConfiguration(new ClientConfiguration()
                .withMaxConnections(producerCount + consumerCount))
        .build();
```

Para `AmazonSQSAsyncClient`, también debe asegurarse de que haya suficientes subprocesos disponibles.
Este ejemplo solo funciona para la versión 1.x de Java.

Agrupación en lotes de acciones

La agrupación por lotes realiza más trabajo durante cada ciclo de ida y vuelta al servicio (por ejemplo, al enviar varios mensajes en una única solicitud `SendMessageBatch`). Las acciones de procesamiento por lotes de Amazon SQS son `SendMessageBatch`, `DeleteMessageBatch` y `ChangeMessageVisibilityBatch`. Para aprovechar el procesamiento por lotes sin modificar los productores ni los consumidores, puede utilizar el Cliente asincrónico en búfer de Amazon SQS.

> **Note**
>
> Debido a que `ReceiveMessage` puede procesar 10 mensajes a la vez, no hay ninguna acción `ReceiveMessageBatch`.

La agrupación en lotes distribuye la latencia de la acción por lotes entre varios mensajes en una solicitud por lotes, en lugar de aceptar toda la latencia para un único mensaje (por ejemplo, una solicitud `SendMessage`). Como en cada ciclo de ida y vuelta se realiza más trabajo, las solicitudes por lotes hacen un uso más eficaz de los subprocesos y las conexiones, por lo que se mejora el desempeño.

Puede combinar la agrupación en lotes con el escalado horizontal para proporcionar un desempeño con menos subprocesos, conexiones y solicitudes de los que serían necesarios en el caso de utilizar solicitudes de mensajes individuales. Puede utilizar acciones de Amazon SQS por lotes para enviar, recibir o eliminar hasta diez mensajes a la vez. Dado que el uso de Amazon SQS se factura por solicitudes, el procesamiento por lotes puede reducir significativamente los costos.

La agrupación en lotes puede introducir cierta complejidad en una aplicación (por ejemplo, la aplicación debe acumular los mensajes antes de enviarlos, o a veces debe esperar más para recibir una respuesta). Sin embargo, la agrupación en lotes puede resultar eficaz en los casos siguientes:

- Cuando la aplicación genera muchos mensajes en poco tiempo, por lo que el retraso nunca es muy largo.
- Cuando un consumidor de mensajes busca mensajes en una cola a discreción, a diferencia de los productores de mensajes típicos que tienen que enviar mensajes como respuesta a eventos que no controlan.

> **Important**
>
> Una solicitud por lotes puede realizarse correctamente aunque se hayan producido errores en mensajes individuales del lote. Después de una solicitud por lotes, compruebe siempre si hay errores en mensajes individuales y vuelva a intentar la acción si es necesario.

Ejemplo funcional en Java de solicitudes de una sola operación y por lotes

Requisitos previos

Añada los paquetes `aws-java-sdk-sqs.jar`, `aws-java-sdk-ec2.jar` y `commons-logging.jar` a la ruta de clases de compilación Java. Los siguientes ejemplos muestran estas dependencias en el archivo `pom.xml` de un proyecto Maven.

```
<dependencies>
    <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-java-sdk-sqs</artifactId>
        <version>LATEST</version>
    </dependency>
    <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-java-sdk-ec2</artifactId>
        <version>LATEST</version>
    </dependency>
    <dependency>
        <groupId>commons-logging</groupId>
        <artifactId>commons-logging</artifactId>
        <version>LATEST</version>
    </dependency>
</dependencies>
```

SimpleProducerConsumer.java

En el siguiente ejemplo de código Java se implementa un patrón productor-consumidor sencillo. El subproceso principal genera una serie de subprocesos productores y consumidores que procesan mensajes de 1 KB durante el tiempo especificado. Este ejemplo incluye productores y consumidores que realizan solicitudes de una única operación y otros que realizan solicitudes por lotes.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

import com.amazonaws.AmazonClientException;
import com.amazonaws.ClientConfiguration;
import com.amazonaws.services.sqs.AmazonSQS;
import com.amazonaws.services.sqs.AmazonSQSClientBuilder;
import com.amazonaws.services.sqs.model.*;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import java.math.BigInteger;
import java.util.ArrayList;
import java.util.List;
import java.util.Random;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.atomic.AtomicBoolean;
import java.util.concurrent.atomic.AtomicInteger;

/**
 * Start a specified number of producer and consumer threads, and produce-consume
 * for the least of the specified duration and 1 hour. Some messages can be left
```

```
 * in the queue because producers and consumers might not be in exact balance.
 */
public class SimpleProducerConsumer {

    // The maximum runtime of the program.
    private final static int MAX_RUNTIME_MINUTES = 60;
    private final static Log log = LogFactory.getLog(SimpleProducerConsumer.class);

    public static void main(String[] args) throws InterruptedException {

        final Scanner input = new Scanner(System.in);

        System.out.print("Enter the queue name: ");
        final String queueName = input.nextLine();

        System.out.print("Enter the number of producers: ");
        final int producerCount = input.nextInt();

        System.out.print("Enter the number of consumers: ");
        final int consumerCount = input.nextInt();

        System.out.print("Enter the number of messages per batch: ");
        final int batchSize = input.nextInt();

        System.out.print("Enter the message size in bytes: ");
        final int messageSizeByte = input.nextInt();

        System.out.print("Enter the run time in minutes: ");
        final int runTimeMinutes = input.nextInt();

        /*
         * Create a new instance of the builder with all defaults (credentials
         * and region) set automatically. For more information, see Creating
         * Service Clients in the AWS SDK for Java Developer Guide.
         */
        final ClientConfiguration clientConfiguration = new ClientConfiguration()
                .withMaxConnections(producerCount + consumerCount);

        final AmazonSQS sqsClient = AmazonSQSClientBuilder.standard()
                .withClientConfiguration(clientConfiguration)
                .build();

        final String queueUrl = sqsClient
                .getQueueUrl(new GetQueueUrlRequest(queueName)).getQueueUrl();
```

```
        // The flag used to stop producer, consumer, and monitor threads.
        final AtomicBoolean stop = new AtomicBoolean(false);

        // Start the producers.
        final AtomicInteger producedCount = new AtomicInteger();
        final Thread[] producers = new Thread[producerCount];
        for (int i = 0; i < producerCount; i++) {
            if (batchSize == 1) {
                producers[i] = new Producer(sqsClient, queueUrl, messageSizeByte,
                        producedCount, stop);
            } else {
                producers[i] = new BatchProducer(sqsClient, queueUrl, batchSize,
                        messageSizeByte, producedCount,
                        stop);
            }
            producers[i].start();
        }

        // Start the consumers.
        final AtomicInteger consumedCount = new AtomicInteger();
        final Thread[] consumers = new Thread[consumerCount];
        for (int i = 0; i < consumerCount; i++) {
            if (batchSize == 1) {
                consumers[i] = new Consumer(sqsClient, queueUrl, consumedCount,
                        stop);
            } else {
                consumers[i] = new BatchConsumer(sqsClient, queueUrl, batchSize,
                        consumedCount, stop);
            }
            consumers[i].start();
        }

        // Start the monitor thread.
        final Thread monitor = new Monitor(producedCount, consumedCount, stop);
        monitor.start();

        // Wait for the specified amount of time then stop.
        Thread.sleep(TimeUnit.MINUTES.toMillis(Math.min(runTimeMinutes,
                MAX_RUNTIME_MINUTES)));
        stop.set(true);

        // Join all threads.
        for (int i = 0; i < producerCount; i++) {
```

```
            producers[i].join();
        }

        for (int i = 0; i < consumerCount; i++) {
            consumers[i].join();
        }

        monitor.interrupt();
        monitor.join();
    }

    private static String makeRandomString(int sizeByte) {
        final byte[] bs = new byte[(int) Math.ceil(sizeByte * 5 / 8)];
        new Random().nextBytes(bs);
        bs[0] = (byte) ((bs[0] | 64) & 127);
        return new BigInteger(bs).toString(32);
    }

    /**
     * The producer thread uses {@code SendMessage}
     * to send messages until it is stopped.
     */
    private static class Producer extends Thread {
        final AmazonSQS sqsClient;
        final String queueUrl;
        final AtomicInteger producedCount;
        final AtomicBoolean stop;
        final String theMessage;

        Producer(AmazonSQS sqsQueueBuffer, String queueUrl, int messageSizeByte,
                 AtomicInteger producedCount, AtomicBoolean stop) {
            this.sqsClient = sqsQueueBuffer;
            this.queueUrl = queueUrl;
            this.producedCount = producedCount;
            this.stop = stop;
            this.theMessage = makeRandomString(messageSizeByte);
        }

        /*
         * The producedCount object tracks the number of messages produced by
         * all producer threads. If there is an error, the program exits the
         * run() method.
         */
        public void run() {
```

```
            try {
                while (!stop.get()) {
                    sqsClient.sendMessage(new SendMessageRequest(queueUrl,
                            theMessage));
                    producedCount.incrementAndGet();
                }
            } catch (AmazonClientException e) {
                /*
                 * By default, AmazonSQSClient retries calls 3 times before
                 * failing. If this unlikely condition occurs, stop.
                 */
                log.error("Producer: " + e.getMessage());
                System.exit(1);
            }
        }
    }

    /**
     * The producer thread uses {@code SendMessageBatch}
     * to send messages until it is stopped.
     */
    private static class BatchProducer extends Thread {
        final AmazonSQS sqsClient;
        final String queueUrl;
        final int batchSize;
        final AtomicInteger producedCount;
        final AtomicBoolean stop;
        final String theMessage;

        BatchProducer(AmazonSQS sqsQueueBuffer, String queueUrl, int batchSize,
                      int messageSizeByte, AtomicInteger producedCount,
                      AtomicBoolean stop) {
            this.sqsClient = sqsQueueBuffer;
            this.queueUrl = queueUrl;
            this.batchSize = batchSize;
            this.producedCount = producedCount;
            this.stop = stop;
            this.theMessage = makeRandomString(messageSizeByte);
        }

        public void run() {
            try {
                while (!stop.get()) {
                    final SendMessageBatchRequest batchRequest =
```

```
                        new SendMessageBatchRequest().withQueueUrl(queueUrl);

                final List<SendMessageBatchRequestEntry> entries =
                        new ArrayList<SendMessageBatchRequestEntry>();
                for (int i = 0; i < batchSize; i++)
                    entries.add(new SendMessageBatchRequestEntry()
                            .withId(Integer.toString(i))
                            .withMessageBody(theMessage));
                batchRequest.setEntries(entries);

                final SendMessageBatchResult batchResult =
                        sqsClient.sendMessageBatch(batchRequest);
                producedCount.addAndGet(batchResult.getSuccessful().size());

                /*
                 * Because SendMessageBatch can return successfully, but
                 * individual batch items fail, retry the failed batch items.
                 */
                if (!batchResult.getFailed().isEmpty()) {
                    log.warn("Producer: retrying sending "
                            + batchResult.getFailed().size() + " messages");
                    for (int i = 0, n = batchResult.getFailed().size();
                         i < n; i++) {
                        sqsClient.sendMessage(new
                                SendMessageRequest(queueUrl, theMessage));
                        producedCount.incrementAndGet();
                    }
                }
            }
        } catch (AmazonClientException e) {
            /*
             * By default, AmazonSQSClient retries calls 3 times before
             * failing. If this unlikely condition occurs, stop.
             */
            log.error("BatchProducer: " + e.getMessage());
            System.exit(1);
        }
    }
}

/**
 * The consumer thread uses {@code ReceiveMessage} and {@code DeleteMessage}
 * to consume messages until it is stopped.
 */
```

```
    private static class Consumer extends Thread {
        final AmazonSQS sqsClient;
        final String queueUrl;
        final AtomicInteger consumedCount;
        final AtomicBoolean stop;

        Consumer(AmazonSQS sqsClient, String queueUrl, AtomicInteger consumedCount,
                 AtomicBoolean stop) {
            this.sqsClient = sqsClient;
            this.queueUrl = queueUrl;
            this.consumedCount = consumedCount;
            this.stop = stop;
        }

        /*
         * Each consumer thread receives and deletes messages until the main
         * thread stops the consumer thread. The consumedCount object tracks the
         * number of messages that are consumed by all consumer threads, and the
         * count is logged periodically.
         */
        public void run() {
            try {
                while (!stop.get()) {
                    try {
                        final ReceiveMessageResult result = sqsClient
                                .receiveMessage(new
                                        ReceiveMessageRequest(queueUrl));

                        if (!result.getMessages().isEmpty()) {
                            final Message m = result.getMessages().get(0);
                            sqsClient.deleteMessage(new
                                    DeleteMessageRequest(queueUrl,
                                    m.getReceiptHandle()));
                            consumedCount.incrementAndGet();
                        }
                    } catch (AmazonClientException e) {
                        log.error(e.getMessage());
                    }
                }
            } catch (AmazonClientException e) {
                /*
                 * By default, AmazonSQSClient retries calls 3 times before
                 * failing. If this unlikely condition occurs, stop.
                 */
```

```
                log.error("Consumer: " + e.getMessage());
                System.exit(1);
            }
        }
    }

    /**
     * The consumer thread uses {@code ReceiveMessage} and {@code
     * DeleteMessageBatch} to consume messages until it is stopped.
     */
    private static class BatchConsumer extends Thread {
        final AmazonSQS sqsClient;
        final String queueUrl;
        final int batchSize;
        final AtomicInteger consumedCount;
        final AtomicBoolean stop;

        BatchConsumer(AmazonSQS sqsClient, String queueUrl, int batchSize,
                      AtomicInteger consumedCount, AtomicBoolean stop) {
            this.sqsClient = sqsClient;
            this.queueUrl = queueUrl;
            this.batchSize = batchSize;
            this.consumedCount = consumedCount;
            this.stop = stop;
        }

        public void run() {
            try {
                while (!stop.get()) {
                    final ReceiveMessageResult result = sqsClient
                            .receiveMessage(new ReceiveMessageRequest(queueUrl)
                                    .withMaxNumberOfMessages(batchSize));

                    if (!result.getMessages().isEmpty()) {
                        final List<Message> messages = result.getMessages();
                        final DeleteMessageBatchRequest batchRequest =
                                new DeleteMessageBatchRequest()
                                        .withQueueUrl(queueUrl);

                        final List<DeleteMessageBatchRequestEntry> entries =
                                new ArrayList<DeleteMessageBatchRequestEntry>();
                        for (int i = 0, n = messages.size(); i < n; i++)
                            entries.add(new DeleteMessageBatchRequestEntry()
                                    .withId(Integer.toString(i))
```

```
                                .withReceiptHandle(messages.get(i)
                                        .getReceiptHandle()));
                        batchRequest.setEntries(entries);

                        final DeleteMessageBatchResult batchResult = sqsClient
                                .deleteMessageBatch(batchRequest);
                        consumedCount.addAndGet(batchResult.getSuccessful().size());

                        /*
                         * Because DeleteMessageBatch can return successfully,
                         * but individual batch items fail, retry the failed
                         * batch items.
                         */
                        if (!batchResult.getFailed().isEmpty()) {
                            final int n = batchResult.getFailed().size();
                            log.warn("Producer: retrying deleting " + n
                                    + " messages");
                            for (BatchResultErrorEntry e : batchResult
                                    .getFailed()) {

                                sqsClient.deleteMessage(
                                        new DeleteMessageRequest(queueUrl,
                                                messages.get(Integer
                                                        .parseInt(e.getId()))
                                                        .getReceiptHandle()));

                                consumedCount.incrementAndGet();
                            }
                        }
                    }
                } catch (AmazonClientException e) {
                    /*
                     * By default, AmazonSQSClient retries calls 3 times before
                     * failing. If this unlikely condition occurs, stop.
                     */
                    log.error("BatchConsumer: " + e.getMessage());
                    System.exit(1);
                }
            }
        }

        /**
         * This thread prints every second the number of messages produced and
```

```
     * consumed so far.
     */
    private static class Monitor extends Thread {
        private final AtomicInteger producedCount;
        private final AtomicInteger consumedCount;
        private final AtomicBoolean stop;

        Monitor(AtomicInteger producedCount, AtomicInteger consumedCount,
                AtomicBoolean stop) {
            this.producedCount = producedCount;
            this.consumedCount = consumedCount;
            this.stop = stop;
        }

        public void run() {
            try {
                while (!stop.get()) {
                    Thread.sleep(1000);
                    log.info("produced messages = " + producedCount.get()
                            + ", consumed messages = " + consumedCount.get());
                }
            } catch (InterruptedException e) {
                // Allow the thread to exit.
            }
        }
    }
}
```

Monitoreo de las métricas de volumen de la ejecución del ejemplo

Amazon SQS genera automáticamente métricas de volumen para los mensajes enviados, recibidos y eliminados. Puedes acceder a esas métricas y a otras a través de la pestaña Supervisión de tu cola o en la CloudWatch consola.

> **Note**
>
> Las métricas pueden tardar en estar disponibles hasta 15 minutos después del inicio de la cola.

Uso de JMS y Amazon SQS

La Biblioteca de mensajes Java de Amazon SQS es una interfaz de Servicio de mensajes de Java (JMS) para Amazon SQS que le permite aprovechar las ventajas de Amazon SQS en aplicaciones que ya utilizan JMS. La interfaz le permite utilizar Amazon SQS como proveedor JMS sin tener que realizar apenas cambios en el código. Junto con AWS SDK for Java, la Biblioteca de mensajes Java de Amazon SQS le permite crear conexiones y sesiones de JMS, así como productores y consumidores que envían y reciben mensajes hacia colas de Amazon SQS y desde ellas.

La biblioteca admite el envío y la recepción de mensajes a una cola (el point-to-point modelo JMS) según la especificación JMS 1.1. La biblioteca permite enviar mensajes de texto, bytes u objetos de forma sincrónica a colas de Amazon SQS. La biblioteca también admite la recepción de objetos de forma sincrónica o asincrónica.

Para obtener más información acerca de las características de la Biblioteca de mensajes Java de Amazon SQS que admiten la especificación JMS 1.1, consulte Amazon SQS admitía implementaciones de JMS 1.1 y las preguntas frecuentes de Amazon SQS.

Temas

Requisitos previos para trabajar con JMS y Amazon SQS

Antes de empezar, debe disponer de los siguientes requisitos previos:

- SDK para Java

  Hay dos formas de incluir el SDK para Java en su proyecto:

  - Descargue e instale el SDK para Java.
  - Utilice Maven para obtener la Biblioteca de mensajes Java de Amazon SQS.

> **Note**
>
> El SDK de Java se incluye como dependencia.
> El SDK para Java y la biblioteca de clientes ampliada de Amazon SQS para Java requieren J2SE Development Kit 8.0 o una versión posterior.

Para obtener información acerca de la descarga de SDK para Java, consulte SDK para Java.

- Biblioteca de mensajes Java de Amazon SQS

  Si no utiliza Maven, debe añadir el paquete `amazon-sqs-java-messaging-lib.jar` a la ruta de clases de Java. Para obtener información sobre la descarga de la biblioteca, consulte Biblioteca de mensajes Java de Amazon SQS.

  > **Note**
  >
  > La Biblioteca de mensajes Java de Amazon SQS incluye compatibilidad con Maven y Spring Framework.
  > Para obtener ejemplos de código en los que se utiliza Maven, el marco Spring y la Biblioteca de mensajes Java de Amazon SQS, consulte Ejemplos prácticos de Java para usar JMS con colas estándar de Amazon SQS.
  >
  > ```
  > <dependency>
  >   <groupId>com.amazonaws</groupId>
  >   <artifactId>amazon-sqs-java-messaging-lib</artifactId>
  >   <version>1.0.4</version>
  >   <type>jar</type>
  > </dependency>
  > ```

- Cola de Amazon SQS

  Cree una cola mediante Amazon SQS, la AWS Management Console API o `CreateQueue` el cliente Amazon SQS empaquetado incluido en la biblioteca de mensajería Java de Amazon SQS.

  - Para obtener información acerca de cómo crear una cola con Amazon SQS a través de la AWS Management Console o la API `CreateQueue`, consulte Creación de colas.
  - Para obtener información acerca del uso de la Biblioteca de mensajes Java de Amazon SQS, consulte Introducción a la Biblioteca de mensajes Java de Amazon SQS.

Introducción a la Biblioteca de mensajes Java de Amazon SQS

Para comenzar a utilizar el Servicio de mensajes de Java (JMS) con Amazon SQS, utilice los ejemplos de código de esta sección. En las próximas secciones se muestra cómo crear una conexión JMS y una sesión y cómo enviar y recibir un mensaje.

El objeto cliente de Amazon SQS encapsulado incluido en la Biblioteca de mensajes Java de Amazon SQS comprueba si existe una cola de Amazon SQS. Si la cola no existe, el cliente la crea.

Creación de una conexión JMS

1. Cree una fábrica de conexiones y llame al método `createConnection` de la fábrica.

   ```
   // Create a new connection factory with all defaults (credentials and region) set
    automatically
   SQSConnectionFactory connectionFactory = new SQSConnectionFactory(
           new ProviderConfiguration(),
           AmazonSQSClientBuilder.defaultClient()
           );

   // Create the connection.
   SQSConnection connection = connectionFactory.createConnection();
   ```

   La clase `SQSConnection` amplía `javax.jms.Connection`. Junto con los métodos de conexión estándar de JMS, `SQSConnection` ofrece métodos adicionales como, por ejemplo, `getAmazonSQSClient` y `getWrappedAmazonSQSClient`. Ambos métodos le permiten llevar a cabo operaciones administrativas no incluidas en la especificación JMS, como la creación de nuevas colas. Sin embargo, el método `getWrappedAmazonSQSClient` también ofrece una versión encapsulada del cliente de Amazon SQS que utiliza la conexión actual. El contenedor transforma cada excepción del cliente en una `JMSException`, por lo que es más fácil que la utilice el código existente, que espera encontrar `JMSException`.
2. Puede utilizar los objetos de cliente devueltos por `getAmazonSQSClient` y `getWrappedAmazonSQSClient` para realizar operaciones administrativas que no están incluidas en la especificación JMS (por ejemplo, puede crear una cola de Amazon SQS).

   Si dispone de código que espera encontrar excepciones de JMS, debería utilizar `getWrappedAmazonSQSClient`:

- Si utiliza `getWrappedAmazonSQSClient`, el objeto de cliente devuelto transforma todas las excepciones en las excepciones de JMS.
- Si utiliza `getAmazonSQSClient`, todas las excepciones serán excepciones de Amazon SQS.

Creación de una cola de Amazon SQS

El objeto de cliente encapsulado comprueba si existe una cola de Amazon SQS.

Si no existe una cola, el cliente la crea. Si la cola existe, la función no devuelve nada. Para obtener más información, consulte la sección "Create the queue if needed" en el ejemplo TextMessageSender.java.

Para crear una cola estándar

```
// Get the wrapped client
AmazonSQSMessagingClientWrapper client = connection.getWrappedAmazonSQSClient();

// Create an SQS queue named MyQueue, if it doesn't already exist
if (!client.queueExists("MyQueue")) {
    client.createQueue("MyQueue");
}
```

Creación de una cola FIFO

```
// Get the wrapped client
AmazonSQSMessagingClientWrapper client = connection.getWrappedAmazonSQSClient();

// Create an Amazon SQS FIFO queue named MyQueue.fifo, if it doesn't already exist
if (!client.queueExists("MyQueue.fifo")) {
    Map<String, String> attributes = new HashMap<String, String>();
    attributes.put("FifoQueue", "true");
    attributes.put("ContentBasedDeduplication", "true");
    client.createQueue(new
 CreateQueueRequest().withQueueName("MyQueue.fifo").withAttributes(attributes));
}
```

> **Note**
>
> La cola FIFO debe finalizar con el sufijo `.fifo`.

Para obtener más información acerca del atributo `ContentBasedDeduplication`, consulte Procesamiento de una sola vez en Amazon SQS.

Envío de mensajes de forma sincrónica

1. Cuando la conexión y la cola de Amazon SQS subyacente estén listas, cree una sesión de JMS sin transacciones con el modo `AUTO_ACKNOWLEDGE`.

```
// Create the nontransacted session with AUTO_ACKNOWLEDGE mode
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
```

2. Para enviar un mensaje de texto a la cola, cree una identidad de cola de JMS y un productor de mensajes.

```
// Create a queue identity and specify the queue name to the session
Queue queue = session.createQueue("MyQueue");

// Create a producer for the 'MyQueue'
MessageProducer producer = session.createProducer(queue);
```

3. Cree un mensaje de texto y envíelo a la cola.

   - Para enviar un mensaje a una cola estándar, no es necesario configurar ningún parámetro adicional.

```
// Create the text message
TextMessage message = session.createTextMessage("Hello World!");

// Send the message
producer.send(message);
System.out.println("JMS Message " + message.getJMSMessageID());
```

   - Para enviar un mensaje a una cola FIFO, debe establecer el ID de grupo de mensajes. También puede establecer un ID de desduplicación de mensajes. Para obtener más información, consulte Términos clave de Amazon SQS.

```
// Create the text message
TextMessage message = session.createTextMessage("Hello World!");

// Set the message group ID
```

```
message.setStringProperty("JMSXGroupID", "Default");

// You can also set a custom message deduplication ID
// message.setStringProperty("JMS_SQS_DeduplicationId", "hello");
// Here, it's not needed because content-based deduplication is enabled for the
 queue

// Send the message
producer.send(message);
System.out.println("JMS Message " + message.getJMSMessageID());
System.out.println("JMS Message Sequence Number " +
 message.getStringProperty("JMS_SQS_SequenceNumber"));
```

Recepción de mensajes de forma sincrónica

1. Para recibir mensajes, cree un consumidor para la misma cola e invoque el método `start`.

   Puede llamar al método `start` de la conexión en cualquier momento. Sin embargo, el consumidor no empieza a recibir mensajes hasta que no se le llama.

   ```
   // Create a consumer for the 'MyQueue'
   MessageConsumer consumer = session.createConsumer(queue);
   // Start receiving incoming messages
   connection.start();
   ```

2. Llame al método `receive` del consumidor con un tiempo de espera establecido en 1 segundo y, a continuación, imprima el contenido del mensaje recibido.

   - Cuando se recibe un mensaje de una cola estándar, se puede obtener acceso al contenido del mensaje.

     ```
     // Receive a message from 'MyQueue' and wait up to 1 second
     Message receivedMessage = consumer.receive(1000);

     // Cast the received message as TextMessage and display the text
     if (receivedMessage != null) {
         System.out.println("Received: " + ((TextMessage) receivedMessage).getText());
     }
     ```

   - Tras recibir un mensaje de una cola FIFO, puede obtener acceso al contenido del mensaje y otros atributos del mensaje específicos de FIFO, como el ID del grupo de mensajes, el ID

de desduplicación de mensajes y el número de secuencia. Para obtener más información, consulte Términos clave de Amazon SQS.

```
// Receive a message from 'MyQueue' and wait up to 1 second
Message receivedMessage = consumer.receive(1000);

// Cast the received message as TextMessage and display the text
if (receivedMessage != null) {
    System.out.println("Received: " + ((TextMessage) receivedMessage).getText());
    System.out.println("Group id: " +
 receivedMessage.getStringProperty("JMSXGroupID"));
    System.out.println("Message deduplication id: " +
 receivedMessage.getStringProperty("JMS_SQS_DeduplicationId"));
    System.out.println("Message sequence number: " +
 receivedMessage.getStringProperty("JMS_SQS_SequenceNumber"));
}
```

3. Cierre la conexión y la sesión.

```
// Close the connection (and the session).
connection.close();
```

El resultado tiene un aspecto similar al siguiente:

```
JMS Message ID:8example-588b-44e5-bbcf-d816example2
Received: Hello World!
```

> **Note**
>
> Puede utilizar el marco Spring para inicializar estos objetos.
> Para obtener más información, consulte `SpringExampleConfiguration.xml`, `SpringExample.java` y el resto de clases auxiliares en `ExampleConfiguration.java` y `ExampleCommon.java` en la sección Ejemplos prácticos de Java para usar JMS con colas estándar de Amazon SQS.

Para obtener ejemplos completos del envío y la recepción de objetos, consulte TextMessageSender.java y SyncMessageReceiver.java.

Recepción de mensajes de forma asincrónica

En el ejemplo de Introducción a la Biblioteca de mensajes Java de Amazon SQS, se envía un mensaje a `MyQueue` y se recibe de forma sincrónica.

El siguiente ejemplo muestra cómo recibir los mensajes de forma asíncrona a través de un agente de escucha.

1. Implemente la interfaz `MessageListener`.

```
class MyListener implements MessageListener {

    @Override
    public void onMessage(Message message) {
        try {
            // Cast the received message as TextMessage and print the text to
 screen.
            System.out.println("Received: " + ((TextMessage) message).getText());
        } catch (JMSException e) {
            e.printStackTrace();
        }
    }
}
```

   Se llama al método onMessage de la interfaz `MessageListener` cuando recibe un mensaje. En esta implementación del agente de escucha, se imprime el texto almacenados en el mensaje.

2. En lugar de llamar explícitamente al método `receive` del consumidor, establezca el agente de escucha de mensajes del consumidor en una instancia de la implementación de `MyListener`. El subproceso principal espera un segundo.

```
// Create a consumer for the 'MyQueue'.
MessageConsumer consumer = session.createConsumer(queue);

// Instantiate and set the message listener for the consumer.
consumer.setMessageListener(new MyListener());

// Start receiving incoming messages.
connection.start();

// Wait for 1 second. The listener onMessage() method is invoked when a message is
 received.
```

```
Thread.sleep(1000);
```

Los pasos restantes son idénticos a los del ejemplo Introducción a la Biblioteca de mensajes Java de Amazon SQS. Para obtener un ejemplo completo de un consumidor asíncrono, consulte `AsyncMessageReceiver.java` Ejemplos prácticos de Java para usar JMS con colas estándar de Amazon SQS.

El resultado de este ejemplo tiene un aspecto similar al siguiente:

```
JMS Message ID:8example-588b-44e5-bbcf-d816example2
Received: Hello World!
```

Uso del modo de reconocimiento del cliente

En el ejemplo de Introducción a la Biblioteca de mensajes Java de Amazon SQS, se utiliza el modo `AUTO_ACKNOWLEDGE`, en el que cada mensaje recibido se confirma automáticamente (y, por tanto, se elimina de la cola subyacente de Amazon SQS).

1. Para reconocer explícitamente los mensajes una vez procesados, debe crear la sesión con el modo `CLIENT_ACKNOWLEDGE`.

   ```
   // Create the non-transacted session with CLIENT_ACKNOWLEDGE mode.
   Session session = connection.createSession(false, Session.CLIENT_ACKNOWLEDGE);
   ```

2. Cuando se reciba el mensaje, muéstrelo y acéptelo explícitamente.

   ```
   // Cast the received message as TextMessage and print the text to screen. Also
    acknowledge the message.
   if (receivedMessage != null) {
       System.out.println("Received: " + ((TextMessage) receivedMessage).getText());
       receivedMessage.acknowledge();
       System.out.println("Acknowledged: " + message.getJMSMessageID());
   }
   ```

   > **Note**
   >
   > En este modo, cuando un mensaje se reconoce, todos los mensajes recibidos antes que este mensaje también se reconocen implícitamente. Por ejemplo, si se reciben 10

mensajes y solo se reconoce el décimo mensaje (en el orden en que los mensajes se reciben), los nueve mensajes anteriores también se reconocen.

Los pasos restantes son idénticos a los del ejemplo Introducción a la Biblioteca de mensajes Java de Amazon SQS. Para ver un ejemplo completo de un consumidor sincrónico con el modo de reconocimiento del cliente, consulte `SyncMessageReceiverClientAcknowledge.java` en Ejemplos prácticos de Java para usar JMS con colas estándar de Amazon SQS.

El resultado de este ejemplo tiene un aspecto similar al siguiente:

```
JMS Message ID:4example-aa0e-403f-b6df-5e02example5
Received: Hello World!
Acknowledged: ID:4example-aa0e-403f-b6df-5e02example5
```

Uso del modo de reconocimiento sin orden

Cuando se utiliza el modo `CLIENT_ACKNOWLEDGE`, todos los mensajes recibidos antes que un mensaje reconocido explícitamente se reconocen automáticamente. Para obtener más información, consulte Uso del modo de reconocimiento del cliente.

La Biblioteca de mensajes Java de Amazon SQS proporciona otro modo de confirmación. Cuando se utiliza el modo `UNORDERED_ACKNOWLEDGE`, el cliente debe reconocer individualmente y de forma explícita todos los mensajes recibidos, independientemente del orden de recepción. Para ello, cree una sesión con el modo `UNORDERED_ACKNOWLEDGE`.

```
// Create the non-transacted session with UNORDERED_ACKNOWLEDGE mode.
Session session = connection.createSession(false, SQSSession.UNORDERED_ACKNOWLEDGE);
```

Los pasos restantes son idénticos a los del ejemplo Uso del modo de reconocimiento del cliente. Para obtener un ejemplo completo de un consumidor síncrono con el modo `UNORDERED_ACKNOWLEDGE`, consulte `SyncMessageReceiverUnorderedAcknowledge.java`.

En este ejemplo, el resultado tiene un aspecto similar al siguiente:

```
JMS Message ID:dexample-73ad-4adb-bc6c-4357example7
Received: Hello World!
Acknowledged: ID:dexample-73ad-4adb-bc6c-4357example7
```

Uso del servicio de mensajes de Java con otros clientes de Amazon SQS

El uso del cliente Java Message Service (JMS) de Amazon SQS con el AWS SDK limita el tamaño de los mensajes de Amazon SQS a 256 KB. No obstante, puede crear un proveedor de JMS mediante cualquier cliente de Amazon SQS. Por ejemplo, puede utilizar el cliente de JMS con la biblioteca de clientes ampliada de Amazon SQS para Java a fin de enviar un mensaje de Amazon SQS que contenga una referencia a una carga de mensaje (de hasta 2 GB) en Amazon S3. Para obtener más información, consulte Administración de mensajes de Amazon SQS de gran tamaño mediante Java y Amazon S3.

En el siguiente ejemplo de código Java se crea el proveedor de JMS para la biblioteca de clientes ampliada:

```
AmazonS3 s3 = new AmazonS3Client(credentials);
Region s3Region = Region.getRegion(Regions.US_WEST_2);
s3.setRegion(s3Region);

// Set the Amazon S3 bucket name, and set a lifecycle rule on the bucket to
// permanently delete objects a certain number of days after each object's creation
 date.
// Next, create the bucket, and enable message objects to be stored in the bucket.
BucketLifecycleConfiguration.Rule expirationRule = new
 BucketLifecycleConfiguration.Rule();
expirationRule.withExpirationInDays(14).withStatus("Enabled");
BucketLifecycleConfiguration lifecycleConfig = new
 BucketLifecycleConfiguration().withRules(expirationRule);

s3.createBucket(s3BucketName);
s3.setBucketLifecycleConfiguration(s3BucketName, lifecycleConfig);
System.out.println("Bucket created and configured.");

// Set the SQS extended client configuration with large payload support enabled.
ExtendedClientConfiguration extendedClientConfig = new ExtendedClientConfiguration()
    .withLargePayloadSupportEnabled(s3, s3BucketName);

AmazonSQS sqsExtended = new AmazonSQSExtendedClient(new AmazonSQSClient(credentials),
 extendedClientConfig);
Region sqsRegion = Region.getRegion(Regions.US_WEST_2);
sqsExtended.setRegion(sqsRegion);
```

El siguiente ejemplo de código de Java crea la fábrica de conexiones:

```
// Create the connection factory using the environment variable credential provider.
// Pass the configured Amazon SQS Extended Client to the JMS connection factory.
SQSConnectionFactory connectionFactory = new SQSConnectionFactory(
        new ProviderConfiguration(),
        sqsExtended
        );

// Create the connection.
SQSConnection connection = connectionFactory.createConnection();
```

Ejemplos prácticos de Java para usar JMS con colas estándar de Amazon SQS

En los siguientes ejemplos de código se muestra cómo utilizar el Servicio de mensajes de Java (JMS) con las colas estándar de Amazon SQS. Para obtener más información sobre cómo trabajar con colas FIFO, consulte Creación de una cola FIFO, Envío de mensajes de forma sincrónica y Recepción de mensajes de forma sincrónica. (La recepción de mensajes de forma sincrónica es la misma para las colas estándar y FIFO. No obstante, los mensajes de las colas FIFO contienen más atributos).

ExampleConfiguration.java

El siguiente ejemplo de código del SDK de Java versión 1.x establece el nombre de la cola por defecto, la región y las credenciales que se utilizarán con los demás ejemplos de Java.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
```

```
 *
 */

public class ExampleConfiguration {
    public static final String DEFAULT_QUEUE_NAME = "SQSJMSClientExampleQueue";

    public static final Region DEFAULT_REGION = Region.getRegion(Regions.US_EAST_2);

    private static String getParameter( String args[], int i ) {
        if( i + 1 >= args.length ) {
            throw new IllegalArgumentException( "Missing parameter for " + args[i] );
        }
        return args[i+1];
    }

    /**
     * Parse the command line and return the resulting config. If the config parsing
 fails
     * print the error and the usage message and then call System.exit
     *
     * @param app the app to use when printing the usage string
     * @param args the command line arguments
     * @return the parsed config
     */
    public static ExampleConfiguration parseConfig(String app, String args[]) {
        try {
            return new ExampleConfiguration(args);
        } catch (IllegalArgumentException e) {
            System.err.println( "ERROR: " + e.getMessage() );
            System.err.println();
            System.err.println( "Usage: " + app + " [--queue <queue>] [--region
 <region>] [--credentials <credentials>] ");
            System.err.println( "  or" );
            System.err.println( "       " + app + " <spring.xml>" );
            System.exit(-1);
            return null;
        }
    }

    private ExampleConfiguration(String args[]) {
        for( int i = 0; i < args.length; ++i ) {
            String arg = args[i];
            if( arg.equals( "--queue" ) ) {
                setQueueName(getParameter(args, i));
```

```
                i++;
            } else if( arg.equals( "--region" ) ) {
                String regionName = getParameter(args, i);
                try {
                    setRegion(Region.getRegion(Regions.fromName(regionName)));
                } catch( IllegalArgumentException e ) {
                    throw new IllegalArgumentException( "Unrecognized region " +
regionName );
                }
                i++;
            } else if( arg.equals( "--credentials" ) ) {
                String credsFile = getParameter(args, i);
                try {
                    setCredentialsProvider( new
PropertiesFileCredentialsProvider(credsFile) );
                } catch (AmazonClientException e) {
                    throw new IllegalArgumentException("Error reading credentials from
" + credsFile, e );
                }
                i++;
            } else {
                throw new IllegalArgumentException("Unrecognized option " + arg);
            }
        }
    }

    private String queueName = DEFAULT_QUEUE_NAME;
    private Region region = DEFAULT_REGION;
    private AWSCredentialsProvider credentialsProvider = new
DefaultAWSCredentialsProviderChain();

    public String getQueueName() {
        return queueName;
    }

    public void setQueueName(String queueName) {
        this.queueName = queueName;
    }

    public Region getRegion() {
        return region;
    }

    public void setRegion(Region region) {
```

```
        this.region = region;
    }

    public AWSCredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }

    public void setCredentialsProvider(AWSCredentialsProvider credentialsProvider) {
        // Make sure they're usable first
        credentialsProvider.getCredentials();
        this.credentialsProvider = credentialsProvider;
    }
}
```

TextMessageSender.java

El siguiente ejemplo de código de Java crea un productor de mensajes de texto.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

public class TextMessageSender {
    public static void main(String args[]) throws JMSException {
        ExampleConfiguration config =
 ExampleConfiguration.parseConfig("TextMessageSender", args);

        ExampleCommon.setupLogging();

        // Create the connection factory based on the config
        SQSConnectionFactory connectionFactory = new SQSConnectionFactory(
```

```
                new ProviderConfiguration(),
                AmazonSQSClientBuilder.standard()
                        .withRegion(config.getRegion().getName())
                        .withCredentials(config.getCredentialsProvider())
                );

        // Create the connection
        SQSConnection connection = connectionFactory.createConnection();

        // Create the queue if needed
        ExampleCommon.ensureQueueExists(connection, config.getQueueName());

        // Create the session
        Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
        MessageProducer producer =
session.createProducer( session.createQueue( config.getQueueName() ) );

        sendMessages(session, producer);

        // Close the connection. This closes the session automatically
        connection.close();
        System.out.println( "Connection closed" );
    }

    private static void sendMessages( Session session, MessageProducer producer ) {
        BufferedReader inputReader = new BufferedReader(
            new InputStreamReader( System.in, Charset.defaultCharset() ) );

        try {
            String input;
            while( true ) {
                System.out.print( "Enter message to send (leave empty to exit): " );
                input = inputReader.readLine();
                if( input == null || input.equals("" ) ) break;

                TextMessage message = session.createTextMessage(input);
                producer.send(message);
                System.out.println( "Send message " + message.getJMSMessageID() );
            }
        } catch (EOFException e) {
            // Just return on EOF
        } catch (IOException e) {
            System.err.println( "Failed reading input: " + e.getMessage() );
        } catch (JMSException e) {
```

```
            System.err.println( "Failed sending message: " + e.getMessage() );
            e.printStackTrace();
        }
    }
}
```

SyncMessageReceiver.java

El siguiente ejemplo de código de Java crea un consumidor de mensajes sincrónico.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

public class SyncMessageReceiver {
public static void main(String args[]) throws JMSException {
    ExampleConfiguration config =
 ExampleConfiguration.parseConfig("SyncMessageReceiver", args);

    ExampleCommon.setupLogging();

    // Create the connection factory based on the config
    SQSConnectionFactory connectionFactory = new SQSConnectionFactory(
            new ProviderConfiguration(),
            AmazonSQSClientBuilder.standard()
                    .withRegion(config.getRegion().getName())
                    .withCredentials(config.getCredentialsProvider())
            );

    // Create the connection
    SQSConnection connection = connectionFactory.createConnection();
```

```
    // Create the queue if needed
    ExampleCommon.ensureQueueExists(connection, config.getQueueName());

    // Create the session
    Session session = connection.createSession(false, Session.CLIENT_ACKNOWLEDGE);
    MessageConsumer consumer =
 session.createConsumer( session.createQueue( config.getQueueName() ) );

    connection.start();

    receiveMessages(session, consumer);

    // Close the connection. This closes the session automatically
    connection.close();
    System.out.println( "Connection closed" );
}

private static void receiveMessages( Session session, MessageConsumer consumer ) {
    try {
        while( true ) {
            System.out.println( "Waiting for messages");
            // Wait 1 minute for a message
            Message message = consumer.receive(TimeUnit.MINUTES.toMillis(1));
            if( message == null ) {
                System.out.println( "Shutting down after 1 minute of silence" );
                break;
            }
            ExampleCommon.handleMessage(message);
            message.acknowledge();
            System.out.println( "Acknowledged message " + message.getJMSMessageID() );
        }
    } catch (JMSException e) {
        System.err.println( "Error receiving from SQS: " + e.getMessage() );
        e.printStackTrace();
    }
}
}
```

AsyncMessageReceiver.java

El siguiente ejemplo de código de Java crea un consumidor de mensajes asincrónico.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

public class AsyncMessageReceiver {
    public static void main(String args[]) throws JMSException, InterruptedException {
        ExampleConfiguration config =
 ExampleConfiguration.parseConfig("AsyncMessageReceiver", args);

        ExampleCommon.setupLogging();

        // Create the connection factory based on the config
        SQSConnectionFactory connectionFactory = new SQSConnectionFactory(
                new ProviderConfiguration(),
                AmazonSQSClientBuilder.standard()
                        .withRegion(config.getRegion().getName())
                        .withCredentials(config.getCredentialsProvider())
                );

        // Create the connection
        SQSConnection connection = connectionFactory.createConnection();

        // Create the queue if needed
        ExampleCommon.ensureQueueExists(connection, config.getQueueName());

        // Create the session
        Session session = connection.createSession(false, Session.CLIENT_ACKNOWLEDGE);
        MessageConsumer consumer =
 session.createConsumer( session.createQueue( config.getQueueName() ) );

        // No messages are processed until this is called
```

```
        connection.start();

        ReceiverCallback callback = new ReceiverCallback();
        consumer.setMessageListener( callback );

        callback.waitForOneMinuteOfSilence();
        System.out.println( "Returning after one minute of silence" );

        // Close the connection. This closes the session automatically
        connection.close();
        System.out.println( "Connection closed" );
    }


    private static class ReceiverCallback implements MessageListener {
        // Used to listen for message silence
        private volatile long timeOfLastMessage = System.nanoTime();

        public void waitForOneMinuteOfSilence() throws InterruptedException {
            for(;;) {
                long timeSinceLastMessage = System.nanoTime() - timeOfLastMessage;
                long remainingTillOneMinuteOfSilence =
                    TimeUnit.MINUTES.toNanos(1) - timeSinceLastMessage;
                if( remainingTillOneMinuteOfSilence < 0 ) {
                    break;
                }
                TimeUnit.NANOSECONDS.sleep(remainingTillOneMinuteOfSilence);
            }
        }


        @Override
        public void onMessage(Message message) {
            try {
                ExampleCommon.handleMessage(message);
                message.acknowledge();
                System.out.println( "Acknowledged message " +
message.getJMSMessageID() );
                timeOfLastMessage = System.nanoTime();
            } catch (JMSException e) {
                System.err.println( "Error processing message: " + e.getMessage() );
                e.printStackTrace();
            }
        }
```

```
    }
}
```

SyncMessageReceiverClientAcknowledge.java

El siguiente ejemplo de código de Java crea un consumidor sincrónico con el modo de reconocimiento del cliente.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

/**
 * An example class to demonstrate the behavior of CLIENT_ACKNOWLEDGE mode for received
 messages. This example
 * complements the example given in {@link SyncMessageReceiverUnorderedAcknowledge} for
 UNORDERED_ACKNOWLEDGE mode.
 *
 * First, a session, a message producer, and a message consumer are created. Then, two
 messages are sent. Next, two messages
 * are received but only the second one is acknowledged. After waiting for the
 visibility time out period, an attempt to
 * receive another message is made. It's shown that no message is returned for this
 attempt since in CLIENT_ACKNOWLEDGE mode,
 * as expected, all the messages prior to the acknowledged messages are also
 acknowledged.
 *
 * This ISN'T the behavior for UNORDERED_ACKNOWLEDGE mode. Please see {@link
 SyncMessageReceiverUnorderedAcknowledge}
 * for an example.
 */
```

```
public class SyncMessageReceiverClientAcknowledge {

    // Visibility time-out for the queue. It must match to the one set for the queue
 for this example to work.
    private static final long TIME_OUT_SECONDS = 1;

    public static void main(String args[]) throws JMSException, InterruptedException {
        // Create the configuration for the example
        ExampleConfiguration config =
 ExampleConfiguration.parseConfig("SyncMessageReceiverClientAcknowledge", args);

        // Setup logging for the example
        ExampleCommon.setupLogging();

        // Create the connection factory based on the config
        SQSConnectionFactory connectionFactory = new SQSConnectionFactory(
                new ProviderConfiguration(),
                AmazonSQSClientBuilder.standard()
                        .withRegion(config.getRegion().getName())
                        .withCredentials(config.getCredentialsProvider())
                );

        // Create the connection
        SQSConnection connection = connectionFactory.createConnection();

        // Create the queue if needed
        ExampleCommon.ensureQueueExists(connection, config.getQueueName());

        // Create the session  with client acknowledge mode
        Session session = connection.createSession(false, Session.CLIENT_ACKNOWLEDGE);

        // Create the producer and consume
        MessageProducer producer =
 session.createProducer(session.createQueue(config.getQueueName()));
        MessageConsumer consumer =
 session.createConsumer(session.createQueue(config.getQueueName()));

        // Open the connection
        connection.start();

        // Send two text messages
        sendMessage(producer, session, "Message 1");
        sendMessage(producer, session, "Message 2");
```

```
        // Receive a message and don't acknowledge it
        receiveMessage(consumer, false);

        // Receive another message and acknowledge it
        receiveMessage(consumer, true);

        // Wait for the visibility time out, so that unacknowledged messages reappear
in the queue
        System.out.println("Waiting for visibility timeout...");
        Thread.sleep(TimeUnit.SECONDS.toMillis(TIME_OUT_SECONDS));

        // Attempt to receive another message and acknowledge it. This results in
receiving no messages since
        // we have acknowledged the second message. Although we didn't explicitly
acknowledge the first message,
        // in the CLIENT_ACKNOWLEDGE mode, all the messages received prior to the
explicitly acknowledged message
        // are also acknowledged. Therefore, we have implicitly acknowledged the first
message.
        receiveMessage(consumer, true);

        // Close the connection. This closes the session automatically
        connection.close();
        System.out.println("Connection closed.");
    }

    /**
     * Sends a message through the producer.
     *
     * @param producer Message producer
     * @param session Session
     * @param messageText Text for the message to be sent
     * @throws JMSException
     */
    private static void sendMessage(MessageProducer producer, Session session, String
messageText) throws JMSException {
        // Create a text message and send it
        producer.send(session.createTextMessage(messageText));
    }

    /**
     * Receives a message through the consumer synchronously with the default timeout
(TIME_OUT_SECONDS).
```

```
     * If a message is received, the message is printed. If no message is received,
 "Queue is empty!" is
     * printed.
     *
     * @param consumer Message consumer
     * @param acknowledge If true and a message is received, the received message is
 acknowledged.
     * @throws JMSException
     */
    private static void receiveMessage(MessageConsumer consumer, boolean acknowledge)
 throws JMSException {
        // Receive a message
        Message message =
 consumer.receive(TimeUnit.SECONDS.toMillis(TIME_OUT_SECONDS));

        if (message == null) {
            System.out.println("Queue is empty!");
        } else {
            // Since this queue has only text messages, cast the message object and
 print the text
            System.out.println("Received: " + ((TextMessage) message).getText());

            // Acknowledge the message if asked
            if (acknowledge) message.acknowledge();
        }
    }
}
```

SyncMessageReceiverUnorderedAcknowledge.java

El siguiente ejemplo de código de Java crea un consumidor sincrónico con el modo de reconocimiento sin orden.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
```

```
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

/**
 * An example class to demonstrate the behavior of UNORDERED_ACKNOWLEDGE mode for
 received messages. This example
 * complements the example given in {@link SyncMessageReceiverClientAcknowledge} for
 CLIENT_ACKNOWLEDGE mode.
 *
 * First, a session, a message producer, and a message consumer are created. Then, two
 messages are sent. Next, two messages
 * are received but only the second one is acknowledged. After waiting for the
 visibility time out period, an attempt to
 * receive another message is made. It's shown that the first message received in the
 prior attempt is returned again
 * for the second attempt. In UNORDERED_ACKNOWLEDGE mode, all the messages must be
 explicitly acknowledged no matter what
 * the order they're received.
 *
 * This ISN'T the behavior for CLIENT_ACKNOWLEDGE mode. Please see {@link
 SyncMessageReceiverClientAcknowledge}
 * for an example.
 */
public class SyncMessageReceiverUnorderedAcknowledge {

    // Visibility time-out for the queue. It must match to the one set for the queue
 for this example to work.
    private static final long TIME_OUT_SECONDS = 1;

    public static void main(String args[]) throws JMSException, InterruptedException {
        // Create the configuration for the example
        ExampleConfiguration config =
 ExampleConfiguration.parseConfig("SyncMessageReceiverUnorderedAcknowledge", args);

        // Setup logging for the example
        ExampleCommon.setupLogging();

        // Create the connection factory based on the config
        SQSConnectionFactory connectionFactory = new SQSConnectionFactory(
                new ProviderConfiguration(),
                AmazonSQSClientBuilder.standard()
```

```
                        .withRegion(config.getRegion().getName())
                        .withCredentials(config.getCredentialsProvider())
                );

        // Create the connection
        SQSConnection connection = connectionFactory.createConnection();

        // Create the queue if needed
        ExampleCommon.ensureQueueExists(connection, config.getQueueName());

        // Create the session  with unordered acknowledge mode
        Session session = connection.createSession(false,
SQSSession.UNORDERED_ACKNOWLEDGE);

        // Create the producer and consume
        MessageProducer producer =
session.createProducer(session.createQueue(config.getQueueName()));
        MessageConsumer consumer =
session.createConsumer(session.createQueue(config.getQueueName()));

        // Open the connection
        connection.start();

        // Send two text messages
        sendMessage(producer, session, "Message 1");
        sendMessage(producer, session, "Message 2");

        // Receive a message and don't acknowledge it
        receiveMessage(consumer, false);

        // Receive another message and acknowledge it
        receiveMessage(consumer, true);

        // Wait for the visibility time out, so that unacknowledged messages reappear
in the queue
        System.out.println("Waiting for visibility timeout...");
        Thread.sleep(TimeUnit.SECONDS.toMillis(TIME_OUT_SECONDS));

        // Attempt to receive another message and acknowledge it. This results in
receiving the first message since
        // we have acknowledged only the second message. In the UNORDERED_ACKNOWLEDGE
mode, all the messages must
        // be explicitly acknowledged.
        receiveMessage(consumer, true);
```

```
        // Close the connection. This closes the session automatically
        connection.close();
        System.out.println("Connection closed.");
    }

    /**
     * Sends a message through the producer.
     *
     * @param producer Message producer
     * @param session Session
     * @param messageText Text for the message to be sent
     * @throws JMSException
     */
    private static void sendMessage(MessageProducer producer, Session session, String
messageText) throws JMSException {
        // Create a text message and send it
        producer.send(session.createTextMessage(messageText));
    }

    /**
     * Receives a message through the consumer synchronously with the default timeout
(TIME_OUT_SECONDS).
     * If a message is received, the message is printed. If no message is received,
"Queue is empty!" is
     * printed.
     *
     * @param consumer Message consumer
     * @param acknowledge If true and a message is received, the received message is
acknowledged.
     * @throws JMSException
     */
    private static void receiveMessage(MessageConsumer consumer, boolean acknowledge)
throws JMSException {
        // Receive a message
        Message message =
consumer.receive(TimeUnit.SECONDS.toMillis(TIME_OUT_SECONDS));

        if (message == null) {
            System.out.println("Queue is empty!");
        } else {
            // Since this queue has only text messages, cast the message object and
print the text
            System.out.println("Received: " + ((TextMessage) message).getText());
```

```
            // Acknowledge the message if asked
            if (acknowledge) message.acknowledge();
        }
    }
}
```

SpringExampleConfiguration.xml

El siguiente ejemplo de código XML es un archivo de configuración bean para SpringExample.java.

```
<!--
    Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.

    Licensed under the Apache License, Version 2.0 (the "License").
    You may not use this file except in compliance with the License.
    A copy of the License is located at

    https://aws.amazon.com/apache2.0

    or in the "license" file accompanying this file. This file is distributed
    on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
    express or implied. See the License for the specific language governing
    permissions and limitations under the License.
-->

<?xml version="1.0" encoding="UTF-8"?>
<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:util="http://www.springframework.org/schema/util"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans http://www.springframework.org/
schema/beans/spring-beans-3.0.xsd
        http://www.springframework.org/schema/util http://www.springframework.org/
schema/util/spring-util-3.0.xsd
    ">

    <bean id="CredentialsProviderBean"
 class="com.amazonaws.auth.DefaultAWSCredentialsProviderChain"/>
```

```
    <bean id="ClientBuilder" class="com.amazonaws.services.sqs.AmazonSQSClientBuilder"
 factory-method="standard">
        <property name="region" value="us-east-2"/>
        <property name="credentials" ref="CredentialsProviderBean"/>
    </bean>

    <bean id="ProviderConfiguration"
 class="com.amazon.sqs.javamessaging.ProviderConfiguration">
        <property name="numberOfMessagesToPrefetch" value="5"/>
    </bean>

    <bean id="ConnectionFactory"
 class="com.amazon.sqs.javamessaging.SQSConnectionFactory">
        <constructor-arg ref="ProviderConfiguration" />
        <constructor-arg ref="ClientBuilder" />
    </bean>

    <bean id="Connection" class="javax.jms.Connection"
        factory-bean="ConnectionFactory"
        factory-method="createConnection"
        init-method="start"
        destroy-method="close" />

    <bean id="QueueName" class="java.lang.String">
        <constructor-arg value="SQSJMSClientExampleQueue"/>
    </bean>
</beans>
```

SpringExample.java

El siguiente ejemplo de código Java utiliza el archivo de configuración bean para inicializar sus objetos.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
```

```
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

public class SpringExample {
    public static void main(String args[]) throws JMSException {
        if( args.length != 1 || !args[0].endsWith(".xml")) {
            System.err.println( "Usage: " + SpringExample.class.getName() + " <spring
 config.xml>" );
            System.exit(1);
        }

        File springFile = new File( args[0] );
        if( !springFile.exists() || !springFile.canRead() ) {
            System.err.println( "File " + args[0] + " doesn't exist or isn't
 readable.");
            System.exit(2);
        }

        ExampleCommon.setupLogging();

        FileSystemXmlApplicationContext context =
            new FileSystemXmlApplicationContext( "file://" +
 springFile.getAbsolutePath() );

        Connection connection;
        try {
            connection = context.getBean(Connection.class);
        } catch( NoSuchBeanDefinitionException e ) {
            System.err.println( "Can't find the JMS connection to use: " +
 e.getMessage() );
            System.exit(3);
            return;
        }

        String queueName;
        try {
            queueName = context.getBean("QueueName", String.class);
        } catch( NoSuchBeanDefinitionException e ) {
            System.err.println( "Can't find the name of the queue to use: " +
 e.getMessage() );
            System.exit(3);
```

```
            return;
        }

        if( connection instanceof SQSConnection ) {
            ExampleCommon.ensureQueueExists( (SQSConnection) connection, queueName );
        }

        // Create the session
        Session session = connection.createSession(false, Session.CLIENT_ACKNOWLEDGE);
        MessageConsumer consumer =
 session.createConsumer( session.createQueue( queueName) );

        receiveMessages(session, consumer);

        // The context can be setup to close the connection for us
        context.close();
        System.out.println( "Context closed" );
    }

    private static void receiveMessages( Session session, MessageConsumer consumer ) {
        try {
            while( true ) {
                System.out.println( "Waiting for messages");
                // Wait 1 minute for a message
                Message message = consumer.receive(TimeUnit.MINUTES.toMillis(1));
                if( message == null ) {
                    System.out.println( "Shutting down after 1 minute of silence" );
                    break;
                }
                ExampleCommon.handleMessage(message);
                message.acknowledge();
                System.out.println( "Acknowledged message" );
            }
        } catch (JMSException e) {
            System.err.println( "Error receiving from SQS: " + e.getMessage() );
            e.printStackTrace();
        }
    }
}
```

ExampleCommon.java

En el siguiente ejemplo de código Java, se comprueba si existe una cola de Amazon SQS y, si no existe, se crea. También incluye ejemplos de código de registro.

```
/*
 * Copyright 2010-2024 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

public class ExampleCommon {
    /**
     * A utility function to check the queue exists and create it if needed. For most
     * use cases this is usually done by an administrator before the application is
 run.
     */
    public static void ensureQueueExists(SQSConnection connection, String queueName)
 throws JMSException {
        AmazonSQSMessagingClientWrapper client =
 connection.getWrappedAmazonSQSClient();

        /**
         * In most cases, you can do this with just a createQueue call, but
 GetQueueUrl
         * (called by queueExists) is a faster operation for the common case where the
 queue
         * already exists. Also many users and roles have permission to call
 GetQueueUrl
         * but don't have permission to call CreateQueue.
         */
        if( !client.queueExists(queueName) ) {
            client.createQueue( queueName );
```

```
        }
    }

    public static void setupLogging() {
        // Setup logging
        BasicConfigurator.configure();
        Logger.getRootLogger().setLevel(Level.WARN);
    }

    public static void handleMessage(Message message) throws JMSException {
        System.out.println( "Got message " + message.getJMSMessageID() );
        System.out.println( "Content: ");
        if( message instanceof TextMessage ) {
            TextMessage txtMessage = ( TextMessage ) message;
            System.out.println( "\t" + txtMessage.getText() );
        } else if( message instanceof BytesMessage ){
            BytesMessage byteMessage = ( BytesMessage ) message;
            // Assume the length fits in an int - SQS only supports sizes up to 256k so
 that
            // should be true
            byte[] bytes = new byte[(int)byteMessage.getBodyLength()];
            byteMessage.readBytes(bytes);
            System.out.println( "\t" +  Base64.encodeAsString( bytes ) );
        } else if( message instanceof ObjectMessage ) {
            ObjectMessage objMessage = (ObjectMessage) message;
            System.out.println( "\t" + objMessage.getObject() );
        }
    }
}
```

Amazon SQS admitía implementaciones de JMS 1.1

La Biblioteca de mensajes Java de Amazon SQS es compatible con las siguientes implementaciones de JMS 1.1. Para obtener más información sobre las características y capacidades compatibles de la Biblioteca de mensajes Java de Amazon SQS, consulte las preguntas frecuentes de Amazon SQS.

Interfaces comunes admitidas

- `Connection`
- `ConnectionFactory`
- `Destination`

- `Session`
- `MessageConsumer`
- `MessageProducer`

Tipos de mensajes admitidos

- `ByteMessage`
- `ObjectMessage`
- `TextMessage`

Modos de confirmación de mensajes admitidos

- `AUTO_ACKNOWLEDGE`
- `CLIENT_ACKNOWLEDGE`
- `DUPS_OK_ACKNOWLEDGE`
- `UNORDERED_ACKNOWLEDGE`

> **Note**
>
> El modo `UNORDERED_ACKNOWLEDGE` no forma parte de la especificación JMS 1.1. Este modo ayuda a Amazon SQS a permitir que un cliente de JMS confirme explícitamente un mensaje.

Propiedades reservadas y encabezados definidos por JMS

Para enviar mensajes

Al enviar mensajes, puede definir los siguientes encabezados y propiedades para cada mensaje:

- `JMSXGroupID` (necesario para las colas FIFO, no permitido para las colas estándar)
- `JMS_SQS_DeduplicationId` (opcional para las colas FIFO, no permitido para las colas estándar)

Después de enviar mensajes, Amazon SQS establece los siguientes encabezados y propiedades para cada mensaje:

- `JMSMessageID`
- `JMS_SQS_SequenceNumber` (solo para colas FIFO)

Para recibir mensajes

Después de recibir mensajes, Amazon SQS establece los siguientes encabezados y propiedades para cada mensaje:

- `JMSDestination`
- `JMSMessageID`
- `JMSRedelivered`
- `JMSXDeliveryCount`
- `JMSXGroupID` (solo para colas FIFO)
- `JMS_SQS_DeduplicationId` (solo para colas FIFO)
- `JMS_SQS_SequenceNumber` (solo para colas FIFO)

Tutoriales de Amazon SQS

En esta sección, encontrará tutoriales que puede utilizar para explorar las características y funcionalidades de Amazon SQS.

Temas

Creación de una cola de Amazon SQS mediante AWS CloudFormation

Puede usar la AWS CloudFormation consola y una plantilla JSON (o YAML) para crear una cola de Amazon SQS. Para obtener más información, consulte Uso de plantillas de AWS CloudFormation y Recurso `AWS::SQS::Queue` en la Guía del usuario de AWS CloudFormation .

Para usar AWS CloudFormation para crear una cola de Amazon SQS.

1. Copie el siguiente código JSON a un archivo denominado `MyQueue.json`. Para crear una cola estándar, omita las propiedades `FifoQueue` y `ContentBasedDeduplication`. Para obtener más información sobre la desduplicación basada en el contenido, consulte Procesamiento de una sola vez en Amazon SQS.

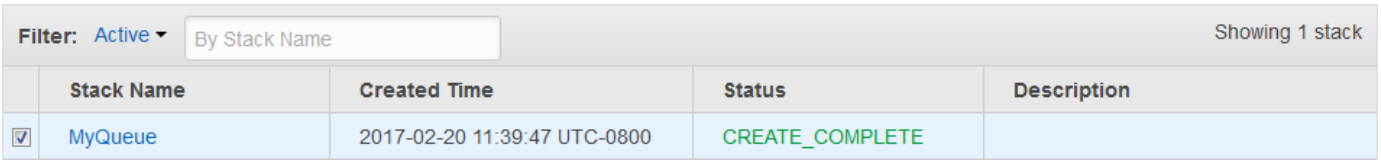
   > **Note**
   >
   > La cola FIFO debe finalizar con el sufijo `.fifo`.

   ```
   {
      "AWSTemplateFormatVersion": "2010-09-09",
      "Resources": {
         "MyQueue": {
            "Properties": {
               "QueueName": "MyQueue.fifo",
               "FifoQueue": true,
               "ContentBasedDeduplication": true
                },
   ```

```
            "Type": "AWS::SQS::Queue"
            }
         },
      "Outputs": {
         "QueueName": {
            "Description": "The name of the queue",
            "Value": {
               "Fn::GetAtt": [
                  "MyQueue",
                  "QueueName"
               ]
            }
         },
         "QueueURL": {
            "Description": "The URL of the queue",
            "Value": {
               "Ref": "MyQueue"
            }
         },
         "QueueARN": {
            "Description": "The ARN of the queue",
            "Value": {
               "Fn::GetAtt": [
                  "MyQueue",
                  "Arn"
               ]
            }
         }
      }
}
```

2. Inicie sesión en la consola de AWS CloudFormation y seleccione Create Stack (Crear pila).
3. En el panel Specify Template (Especificar plantilla), elija Upload a template file (Cargar un archivo de plantilla), elija el archivo `MyQueue.json` y, a continuación, elija Next (Siguiente).
4. En la página Specify Details, escriba `MyQueue` en Stack Name y, a continuación, elija Next.
5. En la página Opciones, seleccione Siguiente.
6. En la página Review (Revisar), elija Create (Crear).

   AWS CloudFormation comienza a crear la `MyQueue` pila y muestra el estado CREATE_IN_PROGRESS. Cuando el proceso se haya completado, AWS CloudFormation mostrará el estado CREATE_COMPLETE.



7. (Opcional) Para mostrar el nombre, la URL y el ARN de la cola, elija el nombre de la pila y, a continuación, en la página siguiente, expanda la sección Outputs.

Tutorial: Envío de un mensaje a una cola de Amazon SQS desde Amazon Virtual Private Cloud

En este tutorial, aprenderá a enviar mensajes a una cola de Amazon SQS a través de una red privada y segura. Esta red consta de una VPC que contiene una instancia de Amazon EC2. La instancia se conecta a Amazon SQS través de un punto de conexión de VPC de interfaz, lo que le permite conectarse a la instancia de Amazon EC2 y enviar mensajes a la cola de Amazon SQS aunque la red esté desconectada del Internet público. Para obtener más información, consulte Puntos de conexión de Amazon Virtual Private Cloud para Amazon SQS.

Important

- Puede usar Amazon Virtual Private Cloud solo con puntos de conexión HTTPS de Amazon SQS.
- Al configurar Amazon SQS para enviar mensajes desde Amazon VPC, debe habilitar el DNS privado y especificar los puntos de conexión en el formato `sqs.us-east-2.amazonaws.com`.
- Los DNS privados no admiten los puntos de enlace heredados, como `queue.amazonaws.com` o `us-east-2.queue.amazonaws.com`.

Temas

Paso 1: Crear un par de claves de Amazon EC2

Los pares de claves permiten conectarse a una instancia de Amazon EC2. Se componen de una clave pública que cifra la información de inicio de sesión y de una clave privada que la descifra.

1. Inicie sesión en la consola de Amazon EC2.
2. En el menú de navegación, en Network & Security (Red y seguridad), seleccione Key Pairs (Pares de claves).
3. Seleccione Create Key Pair.
4. En el cuadro de diálogo Create Key Pair (Crear par de claves), en Key pair name (Nombre del par de claves), escriba `SQS-VPCE-Tutorial-Key-Pair` y haga clic en Create (Crear).
5. El navegador descarga el archivo de clave privada `SQS-VPCE-Tutorial-Key-Pair.pem` automáticamente.

   > **Important**
   >
   > Guarde este archivo en un lugar seguro. EC2 no generará un archivo `.pem` para el mismo par de claves por segunda vez.

6. Para permitir que un cliente SSH pueda conectarse a una instancia EC2, configure los permisos del archivo de clave privada para que solo el usuario pueda leer los permisos; por ejemplo:

   ```
   chmod 400 SQS-VPCE-Tutorial-Key-Pair.pem
   ```

Paso 2: Crear recursos AWS

Para configurar la infraestructura necesaria, debe usar una AWS CloudFormation plantilla, que es un modelo para crear una pila compuesta de AWS recursos, como instancias de Amazon EC2 y colas de Amazon SQS.

La pila de este tutorial incluye los siguientes recursos:

- Una VPC y los recursos de red asociados, como una subred, un grupo de seguridad, una gateway de Internet y una tabla de ruteo.
- Una instancia de Amazon EC2 lanzada en la subred de VPC
- Una cola de Amazon SQS

1. Descargue la plantilla nombrada desde AWS CloudFormation . `SQS-VPCE-Tutorial-CloudFormation.yaml`GitHub
2. Inicie sesión en la consola de AWS CloudFormation.
3. Elija Crear pila.
4. En la página Select Template (Seleccionar plantilla), seleccione Upload a template to Amazon S3 (Cargar una plantilla en Amazon S3), elija el archivo `SQS-VPCE-SQS-Tutorial-CloudFormation.yaml` y haga clic en Next (Siguiente).
5. En la página Specify Details (Especificar detalles), haga lo siguiente:
   a. En Nombre de pila, escriba `SQS-VPCE-Tutorial-Stack`.
   b. Para KeyName, elija SQS-VPCE-Tutorial-Key-Pair.
   c. Elija Siguiente.
6. En la página Opciones, seleccione Siguiente.
7. En la página de revisión, en la sección Capacidades, elija Acepto que AWS CloudFormation podría crear recursos de IAM con nombres personalizados. y, a continuación, elija Crear.

AWS CloudFormation comienza a crear la pila y muestra el estado CREATE_IN_PROGRESS. Cuando el proceso se haya completado, AWS CloudFormation mostrará el estado CREATE_COMPLETE.

Paso 3: confirmar que la instancia EC2 no es de acceso público

La AWS CloudFormation plantilla lanza una instancia EC2 con un nombre `SQS-VPCE-Tutorial-EC2-Instance` en la VPC. Esta instancia de EC2 no permite el tráfico de salida y no puede enviar mensajes a Amazon SQS. Para comprobarlo, debe conectarse a la instancia, intentar conectarse a un punto de conexión público y tratar de enviar un mensaje a Amazon SQS.

1. Inicie sesión en la consola de Amazon EC2.
2. En el menú de navegación, en Instances (Instancias), haga clic en Instances (Instancias).
3. Seleccione SQS-VPCE-. Tutorial-EC2Instance
4. Copie el nombre de host de Public DNS (IPv4); por ejemplo, ec2-203-0-113-0.us-west-2.compute.amazonaws.com.
5. En el directorio que contiene el par de claves que creó anteriormente, conéctese a la instancia utilizando el comando siguiente; por ejemplo:

```
ssh -i SQS-VPCE-Tutorial-Key-Pair.pem ec2-user@ec2-203-0-113-0.us-
east-2.compute.amazonaws.com
```

6. Intente conectarse a un punto de enlace público; por ejemplo:

```
ping amazon.com
```

   Tal y como se esperaba, se produce un error en el intento de conexión.

7. Inicie sesión en la consola de Amazon SQS.
8. En la lista de colas, seleccione la cola creada por la AWS CloudFormation plantilla, por ejemplo, VPCE-SQS-Tutorial-Stack-CFQueue-1abcdefgh2ijk.
9. En la tabla Detalles, copie la URL, por ejemplo, https://sqs.us-east-2.amazonaws.com/123456789012/.
10. En la instancia EC2, intente publicar un mensaje en una cola utilizando el comando siguiente; por ejemplo:

```
aws sqs send-message --region us-east-2 --endpoint-url https://sqs.us-
east-2.amazonaws.com/ --queue-url https://sqs.us-east-2.amazonaws.com/123456789012/
 --message-body "Hello from Amazon SQS."
```

   Tal y como se esperaba, se produce un error al enviar el mensaje.

> **Important**
>
> Posteriormente, cuando cree un punto de conexión de VPC para Amazon SQS, el envío se realizará correctamente.

Paso 4: Crear un punto de conexión de VPC para Amazon SQS

Para conectar su VPC a Amazon SQS, debe definir un punto de conexión de VPC de interfaz. Una vez agregado el punto de conexión, podrá utilizar la API de Amazon SQS desde la instancia de EC2 en su VPC. Esto le permite enviar mensajes a una cola dentro de la AWS red sin cruzar la Internet pública.

> **Note**
>
> La instancia EC2 sigue sin tener acceso a otros AWS servicios y puntos finales de Internet.

1. Inicie sesión en la consola de Amazon VPC.
2. En el menú de navegación, seleccione Endpoints (Puntos de enlace).
3. Seleccione Crear punto de conexión.
4. En la página Crear punto de conexión, en Nombre del servicio, elija el nombre del servicio para Amazon SQS.

   > **Note**
   >
   > Los nombres de los servicios varían en función de la región actual AWS . Por ejemplo, si se encuentra en Este de EE. UU. (Ohio), el nombre del servicio es com.amazonaws.***us-east-2***.sqs.

5. En VPC, seleccione SQS-VPCE-Tutorial-VPC.
6. En Subnets (Subredes), seleccione la red cuyo ID de subred contiene SQS-VPCE-Tutorial-Subnet.
7. En Security group (Grupo de seguridad), haga clic en Select security groups (Seleccionar grupos de seguridad) y elija el grupo de seguridad cuyo nombre de grupo contiene SQS VPCE Tutorial Security Group.
8. Seleccione Crear punto de conexión.

   Se crea el punto de enlace de la VPC de interfaz y se muestra su ID; por ejemplo, vpce-0ab1cdef2ghi3j456k.
9. Elija Close.

   En la consola de Amazon VPC, se abre la página Puntos de enlace.

Amazon VPC comienza a crear el punto de conexión y muestra el estado pendiente. Cuando el proceso se haya completado, Amazon VPC mostrará el estado disponible.

Paso 5: enviar un mensaje a la cola de Amazon SQS

Ahora que la VPC contiene un punto de conexión de Amazon SQS, puede conectarse a la instancia de EC2 y enviar mensajes a la cola.

1. Vuelva a conectarse a la instancia EC2; por ejemplo:

```
ssh -i SQS-VPCE-Tutorial-Key-Pair.pem ec2-user@ec2-203-0-113-0.us-
east-2.compute.amazonaws.com
```

2. Intente publicar de nuevo un mensaje en la cola utilizando el comando siguiente; por ejemplo:

```
aws sqs send-message --region us-east-2 --endpoint-url https://sqs.us-
east-2.amazonaws.com/ --queue-url https://sqs.us-east-2.amazonaws.com/123456789012/
 --message-body "Hello from Amazon SQS."
```

   El intento de envío se realizará correctamente y aparecerán el resumen MD5 del cuerpo del mensaje y el ID del mensaje; por ejemplo:

```
{
 "MD5OfMessageBody": "a1bcd2ef3g45hi678j90klmn12p34qr5",
 "MessageId": "12345a67-8901-2345-bc67-d890123e45fg"
}
```

Para obtener información sobre cómo recibir y eliminar el mensaje de la cola creada por la AWS CloudFormation plantilla (por ejemplo, VPCE-SQS-Tutorial-Stack-CFQueue-1abcdefgh2ijk), consulte. Recibir y eliminar un mensaje en Amazon SQS

Para obtener más información sobre la eliminación de recursos, consulte los siguientes temas:

- Eliminación de un punto de conexión de VPC en la Guía del usuario de Amazon VPC
- Eliminar una cola de Amazon SQS
- Finalice su instancia en la Guía del usuario de Amazon EC2
- Eliminación de la VPC en la Guía del usuario de Amazon VPC
- Eliminar una pila de la AWS CloudFormation consola en la guía del AWS CloudFormation usuario
- Eliminar el par de claves en la Guía del usuario de Amazon EC2

Solución de problemas en Amazon SQS

En los temas siguientes se proporcionan consejos para la resolución de errores y problemas comunes que puede encontrar al utilizar la consola Amazon SQS, la API de Amazon SQS u otras herramientas con Amazon SQS. Si se encuentra con un problema que no aparezca en esta lista, puede utilizar el botón Comentarios de esta página para notificarlo.

Para obtener más consejos sobre la resolución de problemas y respuestas a preguntas comunes de soporte, visite el Centro de conocimientos de AWS.

Temas

Solucionar un error de acceso denegado en Amazon SQS

En los siguientes temas se tratan las causas `AccessDenied` o `AccessDeniedException` errores más comunes en las llamadas a la API de Amazon SQS. Para obtener más información sobre cómo solucionar estos errores, consulte ¿Cómo soluciono los errores de «» o «AccessDeniedAccessDeniedexcepción» en las llamadas a la API de Amazon SQS? en la Guía del Centro de AWS Conocimiento.

Ejemplos de mensajes de error:

```
An error occurred (AccessDenied) when calling the SendMessage operation: Access to
        the resource https://sqs.us-east-1.amazonaws.com/ is denied.
```

- o -

```
An error occurred (KMS.AccessDeniedException) when calling the SendMessage
        operation: User: arn:aws:iam::xxxxx:user/xxxx is not authorized to perform:
        kms:GenerateDataKey on resource: arn:aws:kms:us-east-1:xxxx:key/xxxx with an
 explicit
        deny.
```

Temas

- Política de colas y política de IAM de Amazon SQS
- AWS Key Management Service permisos
- Política de punto de conexión de VPC
- Política de control de servicios de la organización

Política de colas y política de IAM de Amazon SQS

Para comprobar si el solicitante tiene los permisos adecuados para realizar una operación de Amazon SQS, haga lo siguiente:

- Identifique al principal de IAM que realiza la llamada a la API de Amazon SQS. Si el principal de IAM es de la misma cuenta, la política de colas de Amazon SQS o AWS la política de Identity and Access Management (IAM) (Identity and Access Management) deben incluir permisos que permitan explícitamente el acceso a la acción.
- Si el principal es una entidad de IAM:
  - Para identificar a su usuario o rol de IAM, consulte la esquina superior derecha de o utilice el AWS Management Console comando. `aws sts get-caller-identity`
  - Compruebe las políticas de IAM asociadas al usuario o rol de IAM. Puede usar uno de los métodos siguientes:
    - Pruebe las políticas de IAM con el simulador de políticas de IAM.
    - Revisar los diferentes tipos de políticas de IAM.
  - Si es necesario, edite la política de usuario de IAM.
  - Compruebe la política de colas y edítela si es necesario.
- Si el principal es un AWS servicio, la política de colas de Amazon SQS debe permitir el acceso de forma explícita.
- Si el principal es un principal multicuenta, tanto la política de colas de Amazon SQS como la política de IAM deben permitir el acceso de forma explícita.

- Si la política utiliza un elemento de condición, compruebe que la condición restrinja el acceso.

> **Important**
>
> Una denegación explícita en cualquiera de las políticas anula un permiso explícito. Estos son algunos ejemplos básicos de políticas de Amazon SQS.

AWS Key Management Service permisos

Si su cola de Amazon SQS tiene el cifrado del lado del servidor (SSE) activado y una opción gestionada por el cliente AWS KMS key, se deben conceder permisos tanto a los productores como a los consumidores. Para confirmar si una cola está cifrada, puede utilizar el **`KmsMasterKeyId`** atributo de **`GetQueueAttributes`**API o desde la consola de colas, en Cifrado.

- Permisos necesarios para los productores:

```
{
"Effect": "Allow",
"Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
],
"Resource": "<Key ARN>"
}
```

- Permisos necesarios para los consumidores:

```
{
"Effect": "Allow",
"Action": [
    "kms:Decrypt"
],
"Resource": "<Key ARN>"
}
```

- Permisos necesarios para el acceso entre cuentas:

```
{
"Effect": "Allow",
```

```
"Action": [
    "kms:DescribeKey",
    "kms:Decrypt",
    "kms:ReEncrypt",
    "kms:GenerateDataKey"
],
"Resource": "<Key ARN>"
}
```

Puede utilizar cualquiera de las siguientes opciones para habilitar el cifrado de una cola de Amazon SQS:

- SSE-Amazon SQS (clave de cifrado creada y gestionada por el servicio Amazon SQS).
- AWS clave predeterminada administrada (alias/aws/sqs)
- Clave administrada por clientes

Sin embargo, si utiliza una clave KMS AWS administrada, no puede modificar la política de claves predeterminada. Por lo tanto, para proporcionar acceso a otros servicios y cuentas cruzadas, utilice la clave administrada por el cliente. De este modo, podrá editar la política de claves.

Política de punto de conexión de VPC

Si accede a Amazon SQS a través de un punto de enlace de Amazon Virtual Private Cloud (Amazon VPC), la política de punto final de Amazon SQS VPC debe permitir el acceso. Puede crear una política para los puntos de enlace de Amazon VPC para Amazon SQS, donde puede especificar lo siguiente:

1. La entidad principal que puede realizar acciones.
2. Las acciones que se pueden realizar.
3. Los recursos en los que se pueden llevar a cabo las acciones.

En el siguiente ejemplo, la política de puntos finales de la VPC especifica que el usuario *MyUser* de IAM puede enviar mensajes a la cola de Amazon SQS. *MyQueue* A otras acciones, a los usuarios de IAM y a los recursos de Amazon SQS se les niega el acceso a través del punto de enlace de la VPC.

```
{
   "Statement": [{
```

```
      "Action": ["sqs:SendMessage"],
      "Effect": "Allow",
      "Resource": "arn:aws:sqs:us-east-2:123456789012:MyQueue",
      "Principal": {
        "AWS": "arn:aws:iam:123456789012:user/MyUser"
      }
   }]
}
```

Política de control de servicios de la organización

Si Cuenta de AWS pertenece a una organización, AWS Organizations las políticas pueden impedir que acceda a sus colas de Amazon SQS. De forma predeterminada, AWS Organizations las políticas no bloquean ninguna solicitud a Amazon SQS. Sin embargo, asegúrese de que sus AWS Organizations políticas no se hayan configurado para bloquear el acceso a las colas de Amazon SQS. Para obtener instrucciones sobre cómo comprobar sus AWS Organizations políticas, consulte Listar todas las políticas en la Guía del AWS Organizations usuario.

Solucionar problemas de errores de la API de Amazon SQS

En los temas siguientes se describen los errores más comunes que se devuelven al realizar llamadas a la API de Amazon SQS y cómo solucionarlos.

Temas

- QueueDoesNotExist error
- InvalidAttributeValue error
- ReceiptHandle error

QueueDoesNotExist error

Este error se devolverá cuando el servicio Amazon SQS no encuentre la cola mencionada para la acción de Amazon SQS.

Posibles causas y mitigaciones:

- Región incorrecta: revise la configuración del cliente de Amazon SQS para confirmar que ha configurado la región correcta en el cliente. Si no configura una región en el cliente, el SDK o AWS CLI elige la región en el archivo de configuración o en la variable de entorno. Si el SDK no

encuentra ninguna región en el archivo de configuración, establece la región en us-east-1 de forma predeterminada.

- Es posible que la cola se haya eliminado recientemente: si la cola se eliminó antes de realizar la llamada a la API, la llamada a la API devolverá este error. Comprueba si hay CloudTrail alguna DeleteQueueoperación antes de que se produzca el error.
- Problemas con los permisos: si el usuario o rol solicitante AWS Identity and Access Management (de IAM) no tiene los permisos necesarios, es posible que recibas el siguiente error:

```
The specified queue does not exist or you do not have access to it.
```

Comprueba los permisos y realiza la llamada a la API con los permisos correctos.

Para obtener más información sobre cómo solucionar el `QueueDoesNotExist` error, consulte ¿Cómo soluciono el QueueDoesNotExist error cuando realizo llamadas a la API a mi cola de Amazon SQS? en la Guía del Centro de AWS Conocimiento.

InvalidAttributeValue error

Este error se devolverá al actualizar la política de recursos de colas de Amazon SQS o las propiedades con una política o un director incorrectos.

Posibles causas y mitigaciones:

- Política de recursos no válida: compruebe que la política de recursos tenga todos los campos obligatorios. Para obtener más información, consulte la referencia a los elementos de la política JSON de IAM y la validación de las políticas de IAM. También puede usar el generador de políticas de IAM para crear y probar una política de recursos de Amazon SQS. Asegúrese de que la política esté en formato JSON.
- Principal no válido: asegúrese de que el `Principal` elemento existe en la política de recursos y de que el valor es válido. Si su `Principal` elemento de política de recursos de Amazon SQS incluye una entidad de IAM, asegúrese de que la entidad existe antes de usar la política. Amazon SQS valida la política de recursos y comprueba la entidad de IAM. Si la entidad de IAM no existe, recibirá un error. Para confirmar las entidades de IAM, usa las API `GetRole`y `GetUser`.

Para obtener información adicional sobre cómo solucionar un `InvalidAttributeValue` error, consulte ¿Cómo soluciono el QueueDoesNotExist error cuando realizo llamadas a la API a mi cola de Amazon SQS? en la Guía del Centro de Conocimiento AWS .

ReceiptHandle error

Al realizar una llamada a la `DeleteMessage`API, el error `ReceiptHandleIsInvalid` o `InvalidParameterValue` podría devolverse si el identificador del recibo es incorrecto o ha caducado.

- ReceiptHandleIsInvalid error: si el identificador del recibo es incorrecto, recibirás un error similar al de este ejemplo:

```
An error occurred (ReceiptHandleIsInvalid) when calling the DeleteMessage operation:
 The input receipt handle <YOUR RECEIPT HANDLE> is not a valid receipt handle.
```

- InvalidParameterValue error: si el identificador del recibo está caducado, recibirás un error similar al de este ejemplo:

```
An error occurred (InvalidParameterValue) when calling the DeleteMessage operation:
 Value <YOUR RECEIPT HANDLE> for parameter ReceiptHandle is invalid. Reason: The
 receipt handle has expired.
```

Posibles causas y mitigaciones:

El identificador de recepción se crea para cada mensaje recibido y solo es válido durante el período de tiempo de espera de visibilidad. Cuando vence el tiempo de espera de visibilidad, el mensaje pasa a ser visible en la cola para los consumidores. Cuando vuelva a recibir el mensaje del consumidor, recibirá un nuevo identificador de recibo. Para evitar errores en el identificador de los recibos caducados o incorrectos, utiliza el identificador de recibo correcto para eliminar el mensaje dentro del período de espera de visibilidad de las colas de Amazon SQS.

Para obtener información adicional sobre cómo solucionar un `ReceiptHandle` error, consulte ¿Cómo soluciono los errores «» y «ReceiptHandleIsInvalidInvalidParameterValue» cuando utilizo la llamada a la API de Amazon DeleteMessage SQS? en la Guía del Centro de AWS Conocimiento.

Solución de problemas con la cola de cartas muertas de Amazon SQS y los problemas de redrive de DLQ

En los siguientes temas se describen las causas más comunes de los problemas de DLQ y DLQ en Amazon SQS y cómo solucionarlos. Para obtener más información, consulte ¿Cómo se solucionan

los problemas de las reunidades DLQ de Amazon SQS? en la Guía del Centro de Conocimiento AWS .

Temas

- Problemas de DLQ
- Problemas con DLQ-Redrive

Problemas de DLQ

Obtenga información sobre los problemas más comunes de DLQ y cómo resolverlos.

Temas

- La visualización de mensajes mediante la consola puede hacer que los mensajes se muevan a una cola de mensajes fallidos
- Los valores de NumberOfMessagesSent y NumberOfMessagesReceived para una cola de mensajes fallidos no coinciden
- Crear y configurar un redrive de colas con letra muerta
- Gestión de errores de mensajes en cola estándar y FIFO

La visualización de mensajes mediante la consola puede hacer que los mensajes se muevan a una cola de mensajes fallidos

Amazon SQS cuenta la visualización de un mensaje en la consola; para la política de redireccionamiento de la cola correspondiente. Por lo tanto, si ve un mensaje en la consola el número de veces especificado en la política de retransmisión de la cola correspondiente, el mensaje se mueve a la cola de letra muerta de la cola correspondiente.

Para configurar este comportamiento, puede elegir una de las siguientes opciones:

- Aumentar la configuración Maximum Receives de la política de redireccionamiento de la cola correspondiente.
- Evitar visualizar los mensajes de la cola correspondiente en la consola.

Los valores de **`NumberOfMessagesSent`** y **`NumberOfMessagesReceived`** para una cola de mensajes fallidos no coinciden

Si envía manualmente un mensaje a una cola de mensajes fallidos, la métrica `NumberOfMessagesSent` lo captura. Sin embargo, si se envía un mensaje a una cola de mensajes fallidos como consecuencia de un intento fallido de procesamiento, esta métrica no lo captura. Por lo tanto, es posible que los valores de `NumberOfMessagesSent` y `NumberOfMessagesReceived`sean diferentes.

Crear y configurar un redrive de colas con letra muerta

La reactivación de la cola de cartas sin salida requiere que establezca los permisos adecuados para que Amazon SQS reciba mensajes de la cola de cartas sin salida y envíe mensajes a la cola de destino. Si no tiene los permisos correctos, la tarea de reactivar las colas con letra muerta puede fallar. Puedes ver el estado de la tarea de reedición de mensajes para solucionar los problemas y volver a intentarlo.

Gestión de errores de mensajes en cola estándar y FIFO

Las colas estándar siguen procesando los mensajes hasta que finaliza el período de retención. Este procesamiento continuo minimiza las posibilidades de que la cola quede bloqueada por mensajes no consumidos. Tener una gran cantidad de mensajes que el consumidor no puede eliminar repetidamente puede aumentar los costos y suponer una carga adicional para el hardware. Para mantener bajos los costos, transfiera los mensajes fallidos a la cola de letra muerta.

Las colas estándar también permiten enviar un gran número de mensajes durante el vuelo. Si la mayoría de los mensajes no se pueden consumir y no se envían a una cola de espera, el ritmo de procesamiento de los mensajes puede disminuir. Para mantener la eficiencia de la cola, asegúrate de que la aplicación gestione correctamente el procesamiento de los mensajes.

Las colas FIFO proporcionan un procesamiento único al consumir mensajes en secuencia de un grupo de mensajes. Por lo tanto, aunque el consumidor puede seguir recuperando los mensajes ordenados de otro grupo de mensajes, el primer grupo de mensajes no estará disponible hasta que el mensaje que bloquea la cola se procese correctamente o se traslade a una cola de mensajes sin procesar.

Además, las colas FIFO permiten un menor número de mensajes en movimiento. Para evitar que un mensaje bloquee la cola FIFO, asegúrese de que la aplicación gestiona correctamente el procesamiento de los mensajes.

Para obtener más información, consulte Cuotas de mensajes de Amazon SQS y Uso de los mensajes de Amazon SQS.

Problemas con DLQ-Redrive

Obtenga información sobre los problemas más comunes de DLQ-Redrive y cómo resolverlos.

Temas

- AccessDenied problema de permisos
- NonExistentQueue error
- CouldNotDetermineMessageError de origen

AccessDenied problema de permisos

El `AccessDenied` error se produce cuando la reunidad DLQ falla porque la entidad AWS Identity and Access Management (IAM) no tiene los permisos necesarios.

Ejemplo de mensaje de error:

```
Failed to create redrive task. Error code: AccessDenied - Queue Permissions to Redrive.
```

Se requieren los siguientes permisos de API para realizar solicitudes de redrive de DLQ:

Para iniciar una retransmisión de mensajes:

- Permisos de cola con letra muerta:
  - `sqs:StartMessageMoveTask`
  - `sqs:ReceiveMessage`
  - `sqs:DeleteMessage`
  - `sqs:GetQueueAttributes`
  - `kms:Decrypt`— Cuando la cola de letra muerta o la cola de fuentes originales están cifradas.
- Permisos de la cola de destino:
  - `sqs:SendMessage`
  - `kms:GenerateDataKey`— Cuando la cola de destino está cifrada.
  - `kms:Decrypt` — Cuando la cola de destino está cifrada.

Para cancelar una retransmisión de mensajes en curso:

- Permisos de cola con letra muerta:
  - `sqs:CancelMessageMoveTask`
  - `sqs:ReceiveMessage`
  - `sqs:DeleteMessage`
  - `sqs:GetQueueAttributes`
  - `kms:Decrypt`— Cuando la cola de letra muerta o la cola de fuentes originales están cifradas.

Para mostrar el estado de movimiento de un mensaje:

- Permisos de cola con letra muerta:
  - `sqs:ListMessageMoveTasks`
  - `sqs:GetQueueAttributes`

NonExistentQueue error

El `NonExistentQueue` error se produce cuando la cola de origen de Amazon SQS no existe o se ha eliminado. Compruebe y vuelva a conducir hasta una cola de Amazon SQS que esté presente.

Ejemplo de mensaje de error:

```
Failed: AWS.SimpleQueueService.NonExistentQueue
```

CouldNotDetermineMessageError de origen

El `CouldNotDetermineMessageSource` error se produce al intentar iniciar un redrive de DLQ en las siguientes situaciones:

- Un mensaje de Amazon SQS enviado directamente al DLQ con la API. `SendMessage`
- Un mensaje del tema AWS Lambda o función del Amazon Simple Notification Service (Amazon SNS) con el DLQ configurado.

Para resolver este error, selecciona Redrive to a un destino personalizado cuando inicies el redrive. A continuación, introduzca el ARN de la cola de Amazon SQS para mover todos los mensajes del DLQ a la cola de destino.

Ejemplo de mensaje de error:

```
Failed: CouldNotDetermineMessageSource
```

Solución de problemas de regulación de FIFO en Amazon SQS

De forma predeterminada, las colas FIFO admiten 300 transacciones por segundo, por acción de API para `SendMessageReceiveMessage`, y. `DeleteMessage` Las solicitudes de más de 300 TPS reciben el `ThrottlingException` error incluso si los mensajes de la cola están disponibles. Para mitigar esto, puedes usar los siguientes métodos:

- Permita un alto rendimiento para las colas FIFO en Amazon SQS.
- Utilice las acciones `SendMessageBatch` por lotes de la API Amazon SQS y `ChangeMessageVisibilityBatch` aumente el límite de TPS hasta 3000 mensajes por segundo por acción de la API y reduzca los costes. `DeleteMessageBatch` En el caso de la `ReceiveMessage` API, defina el `MaxNumberofMessages` parámetro para recibir hasta diez mensajes por transacción. Para obtener más información, consulte Acciones por lotes de Amazon SQS.
- En el caso de las colas FIFO con un alto rendimiento, siga las recomendaciones para optimizar la utilización de las particiones. Envíe mensajes con los mismos ID de grupo de mensajes en lotes. Elimine los mensajes o cambie los valores de tiempo de espera de visibilidad de los mensajes en lotes con identificadores de recepción procedentes de las mismas `ReceiveMessage` solicitudes de API.
- Aumente el número de valores únicos `MessageGroupId`. Esto permite una distribución uniforme entre las particiones de cola FIFO. Para obtener más información, consulte Uso del ID de grupo de mensajes de Amazon SQS.

Para obtener más información, consulte ¿Por qué mi cola FIFO de Amazon SQS no devuelve todos los mensajes o los mensajes de otros grupos de mensajes? en la Guía del Centro de AWS Conocimiento.

Solucionar problemas de mensajes no devueltos en una llamada a la API de Amazon ReceiveMessage SQS

En los siguientes temas se describen las causas más comunes por las que no se puede devolver un mensaje de Amazon SQS a los consumidores y cómo solucionarlas. Para obtener más información, consulte ¿Por qué no puedo recibir mensajes de mi cola de Amazon SQS? en la guía del AWS Knowledge Center.

Temas

Cola vacía

Para determinar si una cola está vacía, utiliza un sondeo largo para llamar a la `ReceiveMessage`API. También puedes usar las `ApproximateNumberOfMessagesDelayed` CloudWatch métricas `ApproximateNumberOfMessagesVisible`, `ApproximateNumberOfMessagesNotVisible`, y. Si todos los valores de las métricas se establecen en 0 durante varios minutos, la cola se considera vacía.

Se ha alcanzado el límite en vuelo

Si utiliza sondeos prolongados y si se supera el límite de vuelo de la cola (20 000 para el FIFO, 120 000 para el estándar de forma predeterminada), Amazon SQS no devolverá los mensajes de error que superen los límites de la cuota.

Retraso del mensaje

Si la cola de Amazon SQS está configurada como una cola de retraso o los mensajes se enviaron con temporizadores de mensajes, los mensajes no estarán visibles hasta que finalice el tiempo de retraso. Para comprobar si una cola está configurada como cola de retrasos, utilice el **`DelaySeconds`** atributo **`GetQueueAttributes`**API o desde la consola de colas situada en Plazo

de entrega. Comprueba la `ApproximateNumberOfMessagesDelayed` CloudWatch métrica para saber si algún mensaje se retrasa.

El mensaje está en vuelo

Si un consumidor diferente ha sondeado el mensaje, el mensaje estará en movimiento o será invisible durante el tiempo de espera de visibilidad. Es posible que las encuestas adicionales arrojen una recepción vacía. Comprueba la CloudWatch métrica ApproximateNumberOfMessagesVisible para saber el número de mensajes que están disponibles para ser recibidos. En el caso de las colas FIFO, si un mensaje con el identificador del grupo de mensajes está en vuelo, no se devolverán más mensajes a menos que lo elimines o pase a ser visible. Esto se debe a que el orden de los mensajes se mantiene a nivel de grupo de mensajes en una cola FIFO.

Método de sondeo

Si utiliza un sondeo breve (los WaitTimesegundos son 0), Amazon SQS toma muestras de un subconjunto de sus servidores y devuelve los mensajes únicamente de esos servidores. Por lo tanto, es posible que no reciba los mensajes aunque estén disponibles para recibirlos. Las solicitudes de sondeo posteriores devolverán los mensajes.

Si utiliza un sondeo prolongado, Amazon SQS sondea todos los servidores y envía una respuesta después de recopilar al menos un mensaje disponible y hasta el número máximo especificado. Si el valor de ReceiveMessage WaitTimeSegundos es demasiado bajo, es posible que no reciba todos los mensajes disponibles.

Solucionar errores de red en Amazon SQS

En los temas siguientes se describen las causas más comunes de los problemas de red en Amazon SQS y cómo solucionarlos.

Temas

- ETIMEOUT error
- UnknownHostException error

ETIMEOUT error

El ETIMEOUT error se produce cuando el cliente no puede establecer una conexión TCP con un punto final de Amazon SQS.

Solución de problemas:

- Compruebe la conexión de red

  Pruebe su conexión de red a Amazon SQS ejecutando comandos como. `telnet`

  Example: telnet sqs.us-east-1.amazonaws.com 443
- Compruebe la configuración de la red
  - Asegúrese de que las reglas, las rutas y las listas de control de acceso (ACL) del firewall local permitan el tráfico en el puerto que utilice.
  - Las reglas de salida (salida) del grupo de seguridad deben permitir el tráfico al puerto 80 o 443.
  - Las reglas de salida (salida) de la ACL de la red deben permitir el tráfico al puerto TCP 80 o 443.
  - Las reglas de entrada (entrada) de la ACL de la red deben permitir el tráfico en los puertos TCP 1024-65535.
  - Las instancias de Amazon Elastic Compute Cloud (Amazon EC2) que se conectan a la Internet pública deben tener conectividad a Internet.
- Puntos de enlace de Amazon Virtual Private Cloud (Amazon VPC)

  Si accede a Amazon SQS a través de un punto de conexión de Amazon VPC, el grupo de seguridad del punto final debe permitir el tráfico entrante al grupo de seguridad del cliente en el puerto 443. La ACL de red asociada a la subred del punto final de la VPC debe tener esta configuración:
  - Las reglas de salida (egreso) de la ACL de la red deben permitir el tráfico en los puertos TCP 1024-65535 (puertos efímeros).
  - Las reglas de entrada (entrada) de la ACL de la red deben permitir el tráfico en el puerto 443.

Además, la política de puntos finales de VPC AWS Identity and Access Management (IAM) de Amazon SQS debe permitir el acceso. El siguiente ejemplo de política de punto final de VPC especifica que el usuario *`MyUser`*de IAM puede enviar mensajes a la cola de Amazon SQS. *`MyQueue`* A otras acciones, a los usuarios de IAM y a los recursos de Amazon SQS se les niega el acceso a través del punto de enlace de la VPC.

```
{
    "Statement": [{
        "Action": ["sqs:SendMessage"],
```

```
        "Effect": "Allow",
        "Resource": "arn:aws:sqs:us-east-2:123456789012:MyQueue",
        "Principal": {
            "AWS": "arn:aws:iam:123456789012:user/MyUser"
        }
    }]
}
```

UnknownHostException error

El UnknownHostException error se produce cuando no se ha podido determinar la dirección IP del host.

Solución de problemas:

Utilice la nslookup utilidad para devolver la dirección IP asociada al nombre de host:

- Windows and Linux OS

```
nslookup sqs.<region>.amazonaws.com
```

- AWS CLI o SDK para puntos finales heredados de Python:

```
nslookup <region>.queue.amazonaws.com
```

Si has recibido un resultado incorrecto, sigue las instrucciones de ¿Cómo funciona el DNS y cómo se solucionan los errores de DNS parciales o intermitentes? en la Guía del Centro de AWS Conocimiento.

Si ha recibido un resultado válido, es probable que se trate de un problema a nivel de aplicación. Para resolver los problemas a nivel de la aplicación, pruebe los métodos siguientes:

- Reinicie la aplicación.
- Confirme que la aplicación Java no tenga una caché de DNS defectuosa. Si es posible, configure la aplicación para que cumpla con el TTL del DNS. Para obtener más información, consulte Configurar el TTL de la JVM para las búsquedas de nombres DNS.

Para obtener información adicional sobre cómo solucionar errores de red, consulte ¿Cómo soluciono los errores de conexión «ETIMEOUT» y UnknownHost «Exception» de Amazon SQS? en la Guía del Centro de Conocimiento.AWS

Resolución de problemas de las colas de Amazon Simple Queue Service mediante AWS X-Ray

AWS X-Ray recopila datos sobre las solicitudes que atiende su aplicación y le permite ver y filtrar los datos para identificar posibles problemas y oportunidades de optimización. Para cualquier solicitud rastreada hasta su aplicación, puede ver información detallada sobre la solicitud, la respuesta y las llamadas que la aplicación realiza a los AWS recursos intermedios, los microservicios, las bases de datos y las API web HTTP.

Para enviar encabezados de AWS X-Ray seguimiento a través de Amazon SQS, puede realizar una de las siguientes acciones:

- Utilizar el encabezado de rastreo `X-Amzn-Trace-Id`.
- Utilizar el atributo de sistema de mensajes `AWSTraceHeader`.

Para recopilar datos sobre errores y latencia, debe instrumentar el cliente de `AmazonSQS` mediante el SDK de AWS X-Ray.

Puede utilizar la AWS X-Ray consola para ver el mapa de conexiones entre Amazon SQS y otros servicios que utiliza su aplicación. También puede utilizar la consola para ver métricas como la latencia media y las tasas de errores. Para obtener más información, consulte Amazon SQS y AWS X-Ray en la Guía para desarrolladores de AWS X-Ray .

Seguridad en Amazon SQS

En esta sección se proporciona información sobre la seguridad, la autenticación y el control de acceso de Amazon SQS, así como sobre el lenguaje de la política de acceso de Amazon SQS.

Temas

Protección de datos en Amazon SQS

El modelo de se aplica a protección de datos en Amazon Simple Queue Service. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail

- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con Amazon SQS u otro dispositivo Servicios de AWS mediante la consola, la API o AWS los AWS CLI SDK. Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

En las siguientes secciones se proporciona información sobre la protección de datos en Amazon SQS.

Temas

- Cifrado de datos en Amazon SQS
- Privacidad del tráfico entre redes en Amazon SQS

Cifrado de datos en Amazon SQS

La protección de datos se refiere a salvaguardarlos en tránsito (al desplazarse desde y hacia Amazon SQS) y en reposo (almacenados en discos en centros de datos de Amazon SQS). Puede proteger los datos en tránsito con una Capa de sockets seguros (SSL, Secure Sockets Layer) o con el cifrado del lado del cliente. De forma predeterminada, Amazon SQS almacena los mensajes y archivos mediante cifrado de disco. Puede proteger los datos en reposo solicitando a Amazon SQS que cifre sus mensajes antes de guardarlos en el sistema de archivos cifrados de sus centros de datos. Amazon SQS recomienda usar SSE para optimizar el cifrado de datos.

Temas

- Cifrado inactivo en Amazon SQS
- Administración de claves de Amazon SQS

Cifrado inactivo en Amazon SQS

El cifrado del servidor (SSE) le permite transferir información confidencial en colas cifradas. El SSE protege el contenido de los mensajes en las colas mediante claves de cifrado administradas por SQL (SSE-SQS) o claves administradas en el (SSE-KMS). AWS Key Management Service Para obtener información sobre cómo administrar el SSE mediante el, consulte lo siguiente: AWS Management Console

- Configuración de SSE-SQS para una cola (consola)
- Configuración de SSE-KMS para una cola (consola)

Para obtener información sobre la gestión de la ESS mediante las AWS SDK for Java (y `GetQueueAttributes` las acciones `CreateQueueSetQueueAttributes`, y), consulte los siguientes ejemplos:

- Uso del cifrado del lado del servidor con colas de Amazon SQS
- Configurar los permisos de KMS para Servicios de AWS

SSE cifra los mensajes en cuanto Amazon SQS los recibe. Los mensajes se almacenan cifrados y Amazon SQS los descifra únicamente cuando se envían a un consumidor autorizado.

> **Important**
>
> Todas las solicitudes hechas a las colas con SSE habilitado deben usar HTTPS y Signature Version 4.
> Una cola cifrada que usa la clave predeterminada (clave de KMS AWS administrada para Amazon SQS) no puede invocar una función Lambda en otra. Cuenta de AWS
> Algunas funciones de AWS los servicios que pueden enviar notificaciones a Amazon SQS mediante la AWS Security Token Service `AssumeRole` acción son compatibles con SSE, pero solo funcionan con colas estándar:
>
> - Enlaces de ciclo de vida de escalado automático
> - AWS Lambda Colas de mensajes fallidos

Para obtener información acerca de la compatibilidad de otros servicios con temas de cifrado, consulte Configure los permisos de KMS para los servicios AWS y la documentación de los servicios.

AWS KMS combina hardware y software seguros y de alta disponibilidad para proporcionar un sistema de administración de claves adaptado a la nube. Cuando utiliza Amazon SQS con AWS KMS, las claves de datos que cifran los datos de sus mensajes también se cifran y almacenan con los datos que protegen.

A continuación, se describen los beneficios de usar AWS KMS:

- Puede crear y administrar AWS KMS keys usted mismo.
- También puede usar la clave de KMS AWS administrada para Amazon SQS, que es única para cada cuenta y región.
- Los estándares AWS KMS de seguridad pueden ayudarle a cumplir los requisitos de conformidad relacionados con el cifrado.

Para obtener más información, consulte ¿Qué es AWS Key Management Service? en la Guía para desarrolladores de AWS Key Management Service .

Temas

- Ámbito de cifrado
- Términos clave

Ámbito de cifrado

SSE cifra el cuerpo de un mensaje en una cola de Amazon SQS.

SSE no cifra lo siguiente:

- Metadatos de la cola (atributos y nombre de la cola)
- Metadatos del mensaje (ID de mensaje, marca temporal y atributos)
- Métricas por cola

El cifrado de un mensaje evita que usuarios no autorizados o anónimos obtengan acceso a su contenido. Con SSE activado, se rechazarán las solicitudes `SendMessage` y `ReceiveMessage` anónimas a la cola cifrada. Las prácticas recomendadas de seguridad de Amazon SQS desaconsejan utilizar solicitudes anónimas. Si desea enviar solicitudes anónimas a una cola de Amazon SQS, asegúrese de desactivar SSE. Esto no afecta al funcionamiento normal de Amazon SQS:

- Un mensaje se cifra únicamente si se envía con posterioridad a la habilitación del cifrado de una cola. Amazon SQS no cifra mensajes atrasados.
- Cualquier mensaje cifrado permanece en dicho estado aunque el cifrado de su cola esté deshabilitado.

Mover un mensaje a una cola de mensajes fallidos no afecta a su cifrado:

- Cuando Amazon SQS mueve un mensaje de una cola de origen cifrada a una cola de mensajes fallidos sin cifrar, el mensaje permanece cifrado.
- Cuando Amazon SQS mueve un mensaje de una cola de origen sin cifrar a una cola de mensajes fallidos cifrada, el mensaje permanece sin cifrar.

Términos clave

Los siguientes términos clave pueden ayudarle a comprender mejor la funcionalidad de SSE. Para obtener descripciones detalladas, consulte la Referencia de la API de Amazon Simple Queue Service.

Clave de datos

La clave (DEK) es responsable de cifrar el contenido de los mensajes de Amazon SQS.

Para obtener más información, consulte Claves de datos en la Guía para desarrolladores de AWS Key Management Service en la Guía para desarrolladores de AWS Encryption SDK .

Periodo de reutilización de la clave de datos

El tiempo, en segundos, durante el que Amazon SQS puede reutilizar una clave de datos para cifrar o descifrar los mensajes antes de volver a llamar. AWS KMS Un entero que representa segundos, entre 60 segundos (1 minuto) y 86 400 segundos (24 horas). El valor predeterminado es 300 (5 minutos). Para obtener más información, consulte Descripción del período de reutilización de la clave de datos.

> **Note**
>
> En el improbable caso de que no pueda conectarse AWS KMS, Amazon SQS seguirá utilizando la clave de datos en caché hasta que se restablezca la conexión.

ID de clave de KMS

El alias, el ARN del alias, el ID de clave o el ARN de clave de una clave de KMS AWS administrada o una clave de KMS personalizada, en su cuenta o en otra cuenta. Si bien el alias de la clave de KMS AWS administrada para Amazon SQS es siempre`alias/aws/sqs`, el alias de una clave de KMS personalizada puede ser, por ejemplo. `alias/`*`MyAlias`* Puede utilizar estas claves de KMS para proteger los mensajes que se encuentran en las colas de Amazon SQS.

> **Note**
>
> Tenga en cuenta lo siguiente:
>
> - Si no especifica una clave de KMS personalizada, Amazon SQS utilizará la clave de KMS AWS gestionada para Amazon SQS.
> - La primera vez que utilice AWS Management Console para especificar la clave de KMS AWS gestionada de Amazon SQS para una cola, AWS KMS crea la clave de KMS AWS gestionada para Amazon SQS.
> - Como alternativa, la primera vez que utilice la `SendMessageBatch` acción `SendMessage` o en una cola con SSE activado, AWS KMS creará la clave de KMS AWS gestionada para Amazon SQS.

Puede crear claves de KMS, definir las políticas que controlan cómo se pueden usar las claves de KMS y auditar el uso de las claves de KMS mediante la sección de claves administradas por el cliente de la AWS KMS consola o la `CreateKey` AWS KMS acción. Para obtener más información, consulte Claves de KMS y Creación de claves en la Guía para desarrolladores de AWS Key Management Service . Para ver más ejemplos de identificadores clave de KMS, consulta KeyIdla Referencia de la AWS Key Management Service API. Para obtener información sobre la búsqueda de identificadores de claves de KMS, consulte Encontrar el ID y el ARN de la clave en la Guía para desarrolladores de AWS Key Management Service .

> **Important**
>
> Su uso AWS KMS conlleva cargos adicionales. Para obtener más información, consulte Estimación de costos AWS KMS y Precios de AWS Key Management Service.

Cifrado de sobre

La seguridad de los datos cifrados depende en parte de la protección de la clave de datos que permite descifrarlos. Amazon SQS utiliza la clave de KMS para cifrar la clave de datos y, a continuación, la clave de datos cifrada se almacena con el mensaje cifrado. Esta práctica de utilizar una clave de KMS para cifrar las claves de datos se denomina cifrado de sobre.

Para obtener más información, consulte Cifrado de envoltura en la Guía del desarrollador de AWS Encryption SDK .

Administración de claves de Amazon SQS

Amazon SQS se integra con el AWS Key Management Service (KMS) para administrar las claves de KMS para el cifrado del lado del servidor (SSE). Consulte Cifrado inactivo en Amazon SQS para obtener información sobre SSE y las definiciones de administración de claves. Amazon SQS utiliza las claves de KMS para validar y proteger las claves de datos que cifran y descifran los mensajes. En las siguientes secciones se proporciona información sobre cómo trabajar con las claves de KMS y las claves de datos en el servicio de Amazon SQS.

Temas

- Configuración de los permisos de AWS KMS
- Descripción del período de reutilización de la clave de datos
- Estimación de costos AWS KMS
- AWS KMS errores

Configuración de los permisos de AWS KMS

Cada clave de KMS debe tener una política de claves. Tenga en cuenta que no puede modificar la política de claves de una clave de KMS AWS gestionada para Amazon SQS. La política de esta clave de KMS incluye permisos para que todas las entidades principales de la cuenta (que tienen permiso para usar Amazon SQS) utilicen colas cifradas.

Para una clave de KMS administrada por el cliente, debe configurar la política de claves para agregar permisos para cada productor y consumidor de colas. Para ello, designe al productor y al consumidor como usuarios en la política de claves de KMS. Para obtener más información sobre AWS KMS los permisos, consulte la referencia sobre AWS KMS recursos y operaciones o permisos de AWS KMS API en la Guía para AWS Key Management Service desarrolladores.

De forma alternativa, puede especificar los permisos necesarios en una política de IAM asignada a las entidades principales que producen y consumen mensajes cifrados. Para obtener más información, consulte Uso de políticas de IAM con AWS KMS en la Guía para desarrolladores de AWS Key Management Service .

> **Note**
>
> Si bien puede configurar los permisos globales para enviar y recibir desde Amazon SQS, es AWS KMS necesario nombrar explícitamente el ARN completo de las claves de KMS en regiones específicas en la `Resource` sección de una política de IAM.

Configure los permisos de KMS para los servicios AWS

Varios AWS servicios actúan como fuentes de eventos que pueden enviar eventos a las colas de Amazon SQS. Para permitir que estas fuentes de eventos funcionen con colas cifradas, debe crear una clave de KMS administrada por el cliente y añadir permisos en la política de claves para que el servicio utilice los métodos de API necesarios AWS KMS . Realice los siguientes pasos para configurar los permisos.

> **Warning**
>
> Al cambiar la clave de KMS para cifrar los mensajes de Amazon SQS, tenga en cuenta que los mensajes existentes cifrados con la clave de KMS anterior permanecerán cifrados con esa clave. Para descifrar estos mensajes, debe conservar la antigua clave de KMS y asegurarse de que su política de claves concede a Amazon SQS los permisos `kms:Decrypt` para y. `kms:GenerateDataKey` Tras actualizar a una nueva clave de KMS para cifrar los mensajes nuevos, asegúrese de que todos los mensajes existentes cifrados con la antigua clave de KMS se procesen y se eliminen de la cola antes de eliminar o deshabilitar la antigua clave de KMS.

1. Cree una clave de KMS administrada por el cliente. Para obtener más información, consulte Creación de claves en la Guía para desarrolladores de AWS Key Management Service .
2. Para permitir que la fuente AWS de eventos del servicio utilice los métodos `kms:GenerateDataKey` y `kms:Decrypt` API, añada la siguiente declaración a la política de claves de KMS.

```
{
   "Version": "2012-10-17",
      "Statement": [{
         "Effect": "Allow",
         "Principal": {
            "Service": "service.amazonaws.com"
         },
         "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
         ],
         "Resource": "*"
       }]
}
```

   Reemplace "service" (servicio) en el ejemplo anterior por Service name (Nombre del servicio) del origen del evento. Los orígenes de eventos incluyen los siguientes servicios.

Origen del evento	Nombre del servicio
CloudWatch Eventos de Amazon	`events.amazonaws.com`
Notificaciones de eventos de Amazon S3	`s3.amazonaws.com`
Suscripciones a temas de Amazon SNS	`sns.amazonaws.com`

3. Configure una cola SSE existente mediante el ARN de su clave de KMS.
4. Proporcione el ARN de la cola cifrada al origen de eventos.

Configure AWS KMS los permisos para los productores

Cuando caduca el periodo de reutilización de la clave de datos, la siguiente llamada del productor a `SendMessage` o `SendMessageBatch` también desencadena llamadas a `kms:GenerateDataKey`

y `kms:Decrypt`. La llamada a `kms:Decrypt` es para verificar la integridad de la nueva clave de datos antes de usarla. Por lo tanto, el productor debe tener los permisos `kms:GenerateDataKey` y `kms:Decrypt` para la clave de KMS.

Agregue la siguiente instrucción a la política de IAM del productor. Recuerde utilizar los valores de ARN correctos para el recurso de clave y el recurso de cola.

```
{
   "Version": "2012-10-17",
      "Statement": [{
         "Effect": "Allow",
         "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
         ],
         "Resource":  "arn:aws:kms:us-
east-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
         }, {
         "Effect": "Allow",
         "Action": [
            "sqs:SendMessage"
         ],
         "Resource": "arn:aws:sqs:*:123456789012:MyQueue"
      }]
}
```

Configure AWS KMS los permisos para los consumidores

Cuando caduca el periodo de reutilización de la clave de datos, la siguiente llamada del cliente a `ReceiveMessage` también desencadena una llamada a `kms:Decrypt`, para verificar la integridad de la nueva clave de datos antes de usarla. Por lo tanto, el cliente debe tener el permiso `kms:Decrypt` para cualquier clave de KMS que se utilice para cifrar los mensajes en la cola especificada. Si la cola funciona como una cola de mensajes fallidos, el consumidor también debe tener el permiso `kms:Decrypt` para todas las clave de KMS que se utilicen para cifrar los mensajes en la cola de origen. Agregue la siguiente instrucción a la política de IAM del cliente. Recuerde utilizar los valores de ARN correctos para el recurso de clave y el recurso de cola.

```
{
   "Version": "2012-10-17",
      "Statement": [{
         "Effect": "Allow",
```

```
         "Action": [
            "kms:Decrypt"
         ],
         "Resource": "arn:aws:kms:us-
east-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
         }, {
         "Effect": "Allow",
         "Action": [
            "sqs:ReceiveMessage"
         ],
         "Resource": "arn:aws:sqs:*:123456789012:MyQueue"
      }]
}
```

Configure AWS KMS los permisos con una confusa protección adjunta

Cuando la entidad principal de una instrucción de política de claves es una entidad principal de servicio de AWS, puede utilizar las claves de condición global `aws:SourceArn` o `aws:SourceAccount` para protegerse del escenario del suplente confuso. Para utilizar estas claves de condición, establezca el valor al nombre de recurso de Amazon (ARN) o la cuenta del recurso que se está cifrando. Si no conoce el ARN del recurso, utilice `aws:SourceAccount` en su lugar.

En esta política de claves de KMS, un recurso específico de un servicio que sea propiedad de la cuenta `111122223333` puede llamar a KMS para las acciones `Decrypt` y `GenerateDataKey`, que se producen durante el uso de SSE de Amazon SQS.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Principal": {
            "Service": "<replaceable>service</replaceable>.amazonaws.com"
  },
  "Action": [
   "kms:GenerateDataKey",
   "kms:Decrypt"
  ],
  "Resource": "*",
  "Condition": {
   "ArnEquals": {
    "aws:SourceArn": [
     "arn:aws:service::111122223333:resource"
```

```
    ]
   }
  }
 }]
}
```

Cuando se utilizan colas Amazon SQS habilitadas para SSE, los siguientes servicios admiten `aws:SourceArn`:

- Amazon SNS
- Amazon S3
- CloudWatch Eventos
- AWS Lambda
- CodeBuild
- Perfiles de clientes de Amazon Connect
- AWS Auto Scaling
- Amazon Chime

Descripción del período de reutilización de la clave de datos

El periodo de reutilización de la clave de datos define la duración máxima de Amazon SQS para reutilizar la misma clave de datos. Cuando finaliza el periodo de reutilización de la clave de datos, Amazon SQS genera una nueva clave de datos. Tenga en cuenta las siguientes directrices sobre el periodo de reutilización.

- Un período de reutilización más corto proporciona una mayor seguridad, pero se traduce en más llamadas AWS KMS, lo que puede conllevar gastos superiores al nivel gratuito.
- Aunque la clave de datos se almacena en caché de forma independiente para el cifrado y el descifrado, el periodo de reutilización se aplica a ambas copias de la clave de datos.
- Cuando finaliza el período de reutilización de la clave de datos, se realiza la siguiente llamada al método `SendMessage` o `SendMessageBatch` normalmente se activa una llamada al AWS KMS `GenerateDataKey` método para obtener una nueva clave de datos. Además, las siguientes llamadas a, `SendMessage` y cada una de ellas `ReceiveMessage` activará una llamada AWS KMS `Decrypt` a, para comprobar la integridad de la clave de datos antes de utilizarla.
- Los directores (Cuentas de AWS o los usuarios) no comparten claves de datos (los mensajes enviados por directores únicos siempre reciben claves de datos únicas). Por lo tanto, el volumen

de llamadas a AWS KMS es un múltiplo del número de principales únicos que se utilizan durante el período de reutilización de las claves de datos.

Estimación de costos AWS KMS

Para predecir los costes y comprender mejor su AWS factura, quizá le interese saber con qué frecuencia Amazon SQS utiliza su clave de KMS.

> **Note**
>
> Si bien la siguiente fórmula puede brindarle una muy buena idea de los costos esperados, los costos reales podrían ser más elevados debido a la naturaleza distribuida de Amazon SQS.

Para calcular el número de solicitudes de la API (`R`) por cola, utilice la siguiente fórmula:

```
R = (B / D) * (2 * P + C)
```

`B` es el período de facturación (en segundos).

`D` es el período de reutilización de claves de datos (en segundos).

`P` es el número de entidades principales de producción que realizan envíos a la cola de Amazon SQS.

`C` es el número de entidades principales de consumo que reciben información desde la cola de Amazon SQS.

> **Important**
>
> En general, las entidades principales de producción generan el doble del costo que las de consumo. Para obtener más información, consulte Descripción del período de reutilización de la clave de datos.
> Si el productor y el consumidor tienen diferentes usuarios de , el costo aumenta.

A continuación se muestran algunos cálculos de ejemplo. Para obtener información exacta sobre precios, consulte Precios de AWS Key Management Service.

Ejemplo 1: Calcular el número de llamadas a la AWS KMS API para 2 directores y 1 cola

Este ejemplo presupone lo siguiente:

- El período de facturación va del 1 al 31 de enero (2 678 400 segundos).
- El periodo de reutilización de la clave de datos está establecido en 5 minutos (300 segundos).
- Hay una cola.
- Hay una entidad principal de producción y una entidad principal de consumo.

```
(2,678,400 / 300) * (2 * 1 + 1) = 26,784
```

Ejemplo 2: Calcular el número de llamadas a la AWS KMS API para varios productores y consumidores y 2 colas

Este ejemplo presupone lo siguiente:

- El período de facturación va del 1 al 28 de febrero (2 419 200 segundos).
- El periodo de reutilización de la clave de datos está establecido en 24 horas (86 400 segundos).
- Hay 2 colas.
- La primera cola tiene 3 entidades principales productoras y una entidad principal consumidora.
- La segunda cola tiene 5 entidades principales productoras y 2 entidades principales consumidoras.

```
(2,419,200 / 86,400 * (2 * 3 + 1)) + (2,419,200 / 86,400 * (2 * 5 + 2)) = 532
```

AWS KMS errores

Cuando trabaja con Amazon SQS AWS KMS, es posible que se produzcan errores. Las siguientes referencias describen los errores y sus posibles soluciones.

- Errores comunes de AWS KMS
- Errores Decrypt de AWS KMS
- AWS KMS GenerateDataKey errores

Privacidad del tráfico entre redes en Amazon SQS

El punto de conexión de Amazon Virtual Private Cloud (Amazon VPC) para Amazon SQS es una entidad lógica en una VPC que permite la conectividad solo a Amazon SQS. La VPC direcciona las solicitudes a Amazon SQS y vuelve a direccionar las respuestas a la VPC. En las siguientes secciones se proporciona información sobre cómo trabajar con puntos de enlace de la VPC y crear políticas de puntos de enlace de la VPC.

Temas

Puntos de conexión de Amazon Virtual Private Cloud para Amazon SQS

Si utiliza Amazon VPC para alojar sus AWS recursos, puede establecer una conexión entre su VPC y Amazon SQS. Puede utilizar esta conexión para enviar mensajes a sus colas de Amazon SQS sin atravesar el Internet público.

Amazon VPC le permite lanzar AWS recursos en una red virtual personalizada. Puede utilizar una VPC para controlar la configuración de red, como el intervalo de direcciones IP, las subredes, las tablas de enrutamiento y las puertas de enlace de red. Para obtener más información sobre las VPC, consulte la Guía del usuario de Amazon VPC.

Para conectar la VPC a Amazon SQS, primero debe definir un punto de conexión de VPC de interfaz, lo que le permitirá conectar la VPC a otros servicios de AWS . Con el punto de conexión, se ofrece conectividad escalable y fiable con Amazon SQS sin necesidad de utilizar una puerta de enlace de Internet, una instancia de traducción de direcciones de red (NAT) o una conexión de VPN. Para obtener más información, consulte Tutorial: Envío de un mensaje a una cola de Amazon SQS desde Amazon Virtual Private Cloud y Ejemplo 5: denegar el acceso si no es desde un punto de enlace de la VPC en esta guía y Puntos de conexión de VPC de interfaz (AWS PrivateLink) en la Guía del usuario de Amazon VPC.

Important

- Puede usar Amazon Virtual Private Cloud solo con puntos de conexión HTTPS de Amazon SQS.

- Al configurar Amazon SQS para enviar mensajes desde Amazon VPC, debe habilitar el DNS privado y especificar los puntos de conexión en el formato `sqs.us-east-2.amazonaws.com`.
- Los DNS privados no admiten los puntos de enlace heredados, como `queue.amazonaws.com` o `us-east-2.queue.amazonaws.com`.

Creación de una política de punto de conexión de VPC para Amazon SQS

Puede crear una política para los puntos de conexión de Amazon VPC correspondiente a Amazon SQS y especificar lo siguiente:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para obtener más información, consulte Control del acceso a servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC

En la política de puntos de conexión de VPC del ejemplo siguiente, se especifica que el usuario `MyUser` puede enviar mensajes a la cola `MyQueue` de Amazon SQS.

```
{
   "Statement": [{
      "Action": ["sqs:SendMessage"],
      "Effect": "Allow",
      "Resource": "arn:aws:sqs:us-east-2:123456789012:MyQueue",
      "Principal": {
        "AWS": "arn:aws:iam:123456789012:user/MyUser"
      }
   }]
}
```

Se deniega lo siguiente:

- Otras acciones de la API de Amazon SQS, como `sqs:CreateQueue` y `sqs:DeleteQueue`.
- Otros usuarios y reglas de que intentan utilizar este punto de enlace de la VPC.
- El envío de mensajes por parte de `MyUser` a otra cola de Amazon SQS.

> **Note**
>
> El usuario puede seguir utilizando otras acciones de la API de Amazon SQS desde fuera de la VPC. Para obtener más información, consulte Ejemplo 5: denegar el acceso si no es desde un punto de enlace de la VPC.

Identity and Access Management en Amazon SQS

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon SQS. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realice en Amazon SQS.

Usuario de servicio: si utiliza el servicio Amazon SQS para realizar el trabajo, el administrador proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amazon SQS para realizar el trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de Amazon SQS, consulte Solución de problemas de identidad y acceso de Amazon Simple Queue Service.

Administrador de servicio: si está a cargo de los recursos de Amazon SQS de la empresa, probablemente tenga acceso completo a Amazon SQS. El trabajo consiste en determinar a qué características y recursos de Amazon SQS deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información acerca de cómo la empresa puede utilizar IAM con Amazon SQS, consulte Cómo funciona Amazon Simple Queue Service con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que desee obtener información sobre cómo escribir políticas para administrar el acceso a Amazon SQS. Para consultar ejemplos de políticas de Amazon SQS basadas en identidades que puede utilizar en IAM, consulte Prácticas recomendadas sobre las políticas.

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso entre cuentas, consulte el tema sobre el acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos utilizan funciones en otros. Servicios de AWS Servicios de AWS Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar

solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar una AWS función a una instancia EC2 y ponerla a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations .
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado.

Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Información general sobre la administración del acceso en Amazon SQS

Cada AWS recurso es propiedad de un Cuenta de AWS, y los permisos para crear o acceder a un recurso se rigen por las políticas de permisos. Un administrador de la cuenta puede asociar políticas de permisos a identidades de IAM (usuarios, grupos y roles) y algunos servicios (como Amazon SQS) también permiten asociar políticas de permisos a recursos.

> **Note**
>
> Un administrador de la cuenta (o usuario administrador) es un usuario con privilegios administrativos. Para obtener más información, consulte Prácticas recomendadas de IAM en la Guía del usuario de IAM.

Cuando concede permisos, especifica qué usuarios obtienen los permisos, para qué recurso obtienen los permisos y qué acciones específicas desea permitir en el recurso.

Temas

- Recursos y operaciones de Amazon Simple Queue Service
- Titularidad de los recursos
- Administración del acceso a los recursos
- Especificación de elementos de política: acciones, efectos, recursos y entidades principales

Recursos y operaciones de Amazon Simple Queue Service

En Amazon SQS, el único recurso es la cola. En las políticas se emplean nombres de recurso de Amazon (ARN) para identificar los recursos a los que se aplican las políticas. Los siguientes recursos tiene asociado un ARN único:

Tipo de recurso	Formato de ARN
Queue	`arn:aws:sqs:` *`region`*`:`*`account_id`*`:`*`queue_name`*

A continuación, se muestran ejemplos del formato de ARN para las colas:

- Un ARN de una cola denominada `my_queue` en la región EE.UU. Este (Ohio), que pertenece a AWS la cuenta 123456789012:

```
arn:aws:sqs:us-east-2:123456789012:my_queue
```

- Un ARN para una cola denominada `my_queue` en cada una de las diferentes regiones que Amazon SQS admite:

```
arn:aws:sqs:*:123456789012:my_queue
```

- Un ARN que utiliza `*` o `?` como carácter comodín para el nombre de la cola. En los siguientes ejemplos, el ARN coincide con todas las colas que tienen el prefijo `my_prefix_`:

```
arn:aws:sqs:*:123456789012:my_prefix_*
```

Puede obtener el valor de ARN para una cola existente si llama a la acción `GetQueueAttributes`. El valor del atributo `QueueArn` es el ARN de la cola. Para obtener más información sobre los ARN, consulte ARN de IAM en la Guía del usuario de IAM.

Amazon SQS proporciona un conjunto de acciones que funcionan con el recurso de cola. Para obtener más información, consulte Permisos de la API de Amazon SQS: referencia de acciones y recursos.

Titularidad de los recursos

Cuenta de AWS Es propietario de los recursos que se crean en la cuenta, independientemente de quién los haya creado. En concreto, el propietario de los recursos es la Cuenta de AWS de la entidad principal (es decir, la cuenta raíz, un usuario o un rol de IAM) que autentica la solicitud de creación de recursos. Los siguientes ejemplos ilustran cómo funciona:

- Si utiliza las credenciales de su cuenta raíz Cuenta de AWS para crear una cola de Amazon SQS, será el propietario del recurso (en Amazon SQS, el recurso Cuenta de AWS es la cola de Amazon SQS).
- Si crea un usuario en su cuenta Cuenta de AWS y le concede permisos para crear una cola, el usuario podrá crearla. Sin embargo, la propietaria del recurso de cola será su Cuenta de AWS (a la que pertenece el usuario).
- Si crea un rol de IAM en su cuenta Cuenta de AWS con permisos para crear una cola de Amazon SQS, cualquier persona que pueda asumir el rol podrá crear una cola. Usted Cuenta de AWS (al que pertenece el rol) es propietario del recurso de cola.

Administración del acceso a los recursos

Una política de permisos describe los permisos concedidos a las cuentas. En la siguiente sección, se explican las opciones disponibles para crear políticas de permisos.

> **Note**
>
> En esta sección, se explica el uso de IAM en el contexto de Amazon SQS. No se proporciona información detallada sobre el servicio de IAM. Para ver la documentación completa de IAM, consulte What is IAM? (¿Qué es IAM?) en la Guía del usuario de IAM. Para obtener más información acerca de la sintaxis y las descripciones de las políticas de IAM, consulte Referencia de políticas de IAM de AWS en la Guía del usuario de IAM.

Las políticas asociadas a una identidad de IAM se denominan políticas basadas en identidad (políticas de IAM) y las políticas asociadas a un recurso se denominan políticas basadas en recursos.

Políticas basadas en identidad

Hay dos formas de proporcionar a los usuarios permisos para las colas de Amazon SQS: mediante el sistema de políticas de Amazon SQS y mediante el sistema de políticas de IAM. Puede utilizar

cualquiera de estos sistemas, o ambos, para asociar políticas a usuarios o roles. En la mayoría de los casos, puede conseguir el mismo resultado mediante cualquiera de estos dos sistemas. Por ejemplo, puede hacer lo siguiente:

- Adjuntar una política de permisos a un usuario o un grupo de su cuenta: para conceder a un usuario permisos para crear una cola de Amazon SQS, adjunte una política de permisos a un usuario o a un grupo al que este pertenezca.
- Adjuntar una política de permisos a un usuario de otra Cuenta de AWS: para conceder a un usuario permisos para crear una cola de Amazon SQS, adjunte una política de permisos de Amazon SQS a un usuario de otra Cuenta de AWS.

  Los permisos entre cuentas no se aplican a las siguientes acciones:

  - `AddPermission`
  - `CancelMessageMoveTask`
  - `CreateQueue`
  - `DeleteQueue`
  - `ListMessageMoveTask`
  - `ListQueues`
  - `ListQueueTags`
  - `RemovePermission`
  - `SetQueueAttributes`
  - `StartMessageMoveTask`
  - `TagQueue`
  - `UntagQueue`
- Asociar una política de permisos a un rol (conceder permisos entre cuentas): para conceder permisos entre cuentas, adjunte una política de permisos basada en identidad a un rol de IAM. Por ejemplo, el administrador Cuenta de AWS A puede crear un rol para conceder permisos entre cuentas a Cuenta de AWS B (o a un AWS servicio) de la siguiente manera:
  - El administrador de ca Cuenta A crea un rol de IAM y adjunta una política de permisos al rol que concede permisos sobre los recursos de la cuenta A.
  - El administrador de la cuenta A asocia una política de confianza al rol que identifica la cuenta B como la entidad principal que puede asumir el rol.

- El administrador de la cuenta B delega el permiso para asumir el rol a todos los usuarios de la cuenta B. Esto permite a los usuarios de la cuenta B crear u obtener acceso a colas de la cuenta A.

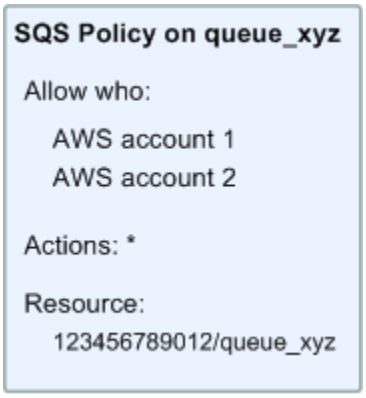
> **Note**
>
> Si quieres conceder el permiso para asumir la función a un AWS servicio, el principal de la política de confianza también puede ser el principal de AWS servicio.

Para obtener más información sobre el uso de IAM para delegar permisos, consulte Administración de accesos en la Guía del usuario de IAM.

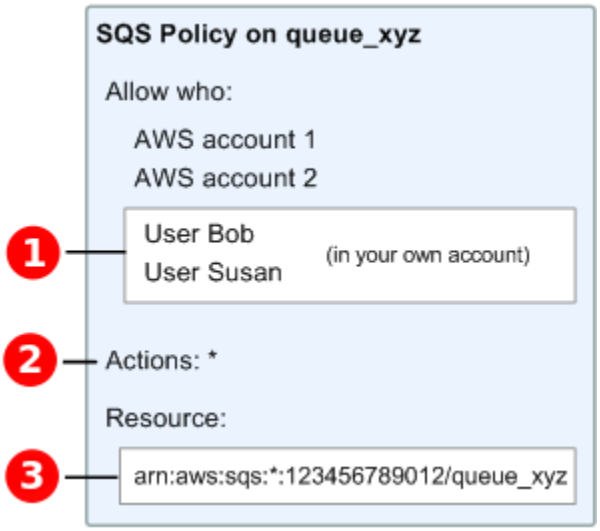
Aunque Amazon SQS funciona con las políticas de IAM tiene su propia infraestructura de políticas. Puede utilizar una política de Amazon SQS con una cola para especificar qué AWS cuentas tienen acceso a la cola. Puede especificar el tipo de acceso y las condiciones (por ejemplo, una condición que concede permisos para utilizar `SendMessage`, `ReceiveMessage` si la solicitud se realiza antes del 31 de diciembre de 2010). Las acciones específicas para las que puede conceder permisos son un subconjunto de la lista general de acciones de Amazon SQS. Cuando se escribe una política de Amazon SQS y se especifica * para “permitir todas las acciones de Amazon SQS”, significa que un usuario puede realizar todas las acciones de este subconjunto.

En el siguiente diagrama, se ilustra el concepto de una de estas políticas básicas de Amazon SQS, que abarca el subconjunto de acciones. La política es válida y otorga permisos a la AWS cuenta 1 y a la AWS cuenta 2 para usar cualquiera de las acciones permitidas con la cola especificada. `queue_xyz`

> **Note**
>
> El recurso de la política se especifica como`123456789012/queue_xyz`, donde `123456789012` está el ID de AWS cuenta de la cuenta propietaria de la cola.



Con la introducción de IAM y los conceptos de usuarios y nombres de recursos de Amazon (ARN), han cambiado algunas cosas con respecto a las políticas de SQS. En el diagrama y la tabla siguientes se describen estos cambios.



1 Para obtener información sobre cómo conceder permisos a los usuarios de distintas cuentas, consulte el tutorial: Delegar el acceso entre AWS cuentas mediante funciones de IAM en la Guía del usuario de IAM.

2 El subconjunto de acciones incluidas en * se ha ampliado. Para obtener una lista de las acciones permitidas, consulte Permisos de la API de Amazon SQS: referencia de acciones y recursos.

3 Puede especificar el recurso mediante el nombre de recurso de Amazon (ARN), que es la forma estándar de especificar recursos en las políticas de IAM. Para obtener información acerca del

formato de ARN para las colas de Amazon SQS, consulte Recursos y operaciones de Amazon Simple Queue Service.

Por ejemplo, de acuerdo con la política de Amazon SQS del diagrama anterior, cualquier persona que posea las credenciales de seguridad de la AWS cuenta 1 o la AWS cuenta 2 puede acceder. `queue_xyz` Además, los usuarios Bob y Susan de su propia cuenta de AWS (con el ID `123456789012`) pueden obtener acceso a la cola.

Antes de la introducción de IAM, Amazon SQS concedía automáticamente al creador de una cola control total sobre esa cola (es decir, acceso a todas las posibles acciones de Amazon SQS en dicha cola). Esto ha cambiado, a menos que el creador utilice credenciales de seguridad de AWS . Cualquier usuario que tenga permisos para crear una cola debe contar también con permisos para utilizar otras acciones de Amazon SQS para poder realizar alguna acción con las colas creadas.

A continuación, se muestra una política de ejemplo que permite a un usuario utilizar todas las acciones de Amazon SQS, pero solo con las colas cuyos nombres tengan como prefijo la cadena literal `bob_queue_`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:*:123456789012:bob_queue_*"
   }]
}
```

Para obtener más información, consulte Uso de políticas con Amazon SQS e Identidades (usuarios, grupos y roles) en la Guía del usuario de IAM.

Especificación de elementos de política: acciones, efectos, recursos y entidades principales

Para cada recurso de Amazon Simple Queue Service, el servicio define un conjunto de acciones. Para conceder permisos a estas acciones, Amazon SQS define un conjunto de acciones que se pueden especificar en una política.

> **Note**
>
> Para realizar una acción de la , pueden ser necesarios permisos para más de una acción. Cuando se conceden permisos para acciones específicas, también debe identificar el recurso para el que las acciones se autorizan o deniegan.

A continuación se indican los elementos más básicos de la política:

- Recurso: en una política, se usa un nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política.
- Acción: utilice palabras de clave de acción para identificar las acciones de recursos que desea permitir o denegar. Por ejemplo, cuando se concede el permiso `sqs:CreateQueue`, el usuario puede realizar la acción `CreateQueue` de Amazon Simple Queue Service.
- Efecto: especifique el efecto que se producirá cuando el usuario solicite la acción específica; puede ser permitir o denegar. Si no concede acceso de forma explícita a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso para asegurarse de que un usuario no pueda obtener acceso a él, aunque otra política le conceda acceso.
- Entidad principal: en las políticas basadas en identidades (políticas de IAM), el usuario al que se asocia esta política es la entidad principal implícita. Para las políticas basadas en recursos, debe especificar el usuario, la cuenta, el servicio u otra entidad que desee que reciba permisos (se aplica solo a las políticas basadas en recursos).

Para obtener más información sobre la sintaxis y descripciones de las políticas de Amazon SQS, consulte Referencia de la política de AWS IAM en la Guía del usuario de IAM.

Para ver una tabla con todas las acciones de Amazon Simple Queue Service y los recursos a los que se aplican, consulte Permisos de la API de Amazon SQS: referencia de acciones y recursos.

Cómo funciona Amazon Simple Queue Service con IAM

Antes de utilizar IAM para administrar el acceso a Amazon SQS, obtenga información sobre qué características de IAM se encuentran disponibles con Amazon SQS.

Características de IAM que puede utilizar con Amazon Simple Queue Service

Característica de IAM	Compatibilidad con Amazon SQS
Políticas basadas en identidades	Sí
Políticas basadas en recursos	Sí
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACL	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Sesiones de acceso directo (FAS)	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan Amazon SQS y otros AWS servicios con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Control de acceso

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

> **Note**
>
> Es importante entender que todos Cuentas de AWS pueden delegar sus permisos a los usuarios en sus cuentas. El acceso multicuenta te permite compartir el acceso a tus AWS recursos sin tener que gestionar usuarios adicionales. Para obtener información sobre el uso del acceso entre cuentas, consulte Habilitación del acceso entre cuentas en la Guía del usuario de IAM.
> Consulte Limitaciones de las políticas personalizadas de Amazon SQS para obtener más detalles sobre los permisos de contenido cruzado y las claves de condición en las políticas personalizadas de Amazon SQS.

Políticas basadas en identidad para Amazon SQS

Compatibilidad con las políticas basadas en identidad	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidad para Amazon SQS

Para ver ejemplos de políticas basadas en identidad de Amazon SQS, consulte Prácticas recomendadas sobre las políticas.

Políticas basadas en recursos de Amazon SQS

Compatibilidad con las políticas basadas en recursos	Sí

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte el tema Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Acciones de políticas para Amazon SQS

Admite acciones de política	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no

tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Amazon SQS, consulte Recursos definidos por Amazon Simple Queue Service en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Amazon SQS utilizan el siguiente prefijo antes de la acción:

```
sqs
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "sqs:action1",
      "sqs:action2"
         ]
```

Para ver ejemplos de políticas basadas en identidad de Amazon SQS, consulte Prácticas recomendadas sobre las políticas.

Recursos de políticas para Amazon SQS

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de los tipos de recursos de Amazon SQS y los ARN, consulte Acciones definidas por Amazon Simple Queue Service en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Recursos definidos por Amazon Simple Queue Service.

Para ver ejemplos de políticas basadas en identidad de Amazon SQS, consulte Prácticas recomendadas sobre las políticas.

Claves de condición de políticas para Amazon SQS

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para obtener una lista de las claves de condición de Amazon SQS, consulte Claves de condición para Amazon Simple Queue Service en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Recursos definidos por Amazon Simple Queue Service.

Para ver ejemplos de políticas basadas en identidad de Amazon SQS, consulte Prácticas recomendadas sobre las políticas.

ACL en Amazon SQS

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con Amazon SQS

Admite ABAC (etiquetas en las políticas)	Parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con Amazon SQS

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Sesiones de acceso directo para Amazon SQS

Admite Forward access sessions (FAS)	Sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones

con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para Amazon SQS

Compatible con roles de servicio	Sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos para un rol de servicio podría interrumpir la funcionalidad de Amazon SQS. Edite los roles de servicio solo cuando Amazon SQS proporcione orientación para hacerlo.

Roles vinculados a servicios para Amazon SQS

Compatible con roles vinculados al servicio	No

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Amazon SQS actualiza las políticas gestionadas AWS

Para agregar permisos a usuarios, grupos y roles, es más fácil utilizar las políticas administradas de AWS que escribirlas uno mismo. Se necesita tiempo y experiencia para crear políticas administradas por el cliente de IAM que le brinden a su equipo solo los permisos necesarios. Para comenzar a hacerlo con rapidez, puede utilizar nuestras políticas administradas de AWS . Estas políticas cubren casos de uso comunes y están disponibles en su cuenta de AWS . Para obtener más información sobre las políticas AWS administradas, consulte las políticas AWS administradas en la Guía del usuario de IAM.

AWS los servicios mantienen y AWS actualizan las políticas gestionadas. No puede cambiar los permisos en las políticas AWS gestionadas. En ocasiones, los servicios añaden permisos adicionales a una política AWS gestionada para admitir nuevas funciones. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Lo más probable es que los servicios actualicen una política AWS administrada cuando se lanza una nueva función o cuando hay nuevas operaciones disponibles. Los servicios no eliminan los permisos de una política AWS administrada, por lo que las actualizaciones de la política no afectarán a los permisos existentes.

Además, AWS admite políticas administradas para funciones laborales que abarcan varios servicios. Por ejemplo, la política de ReadOnlyacceso AWS gestionado proporciona acceso de solo lectura a todos los AWS servicios y recursos. Cuando un servicio lanza una nueva función, AWS agrega permisos de solo lectura para nuevas operaciones y recursos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte Políticas administradas de AWS para funciones de trabajo en la Guía del usuario de IAM.

AWS política gestionada: AmazonSQS FullAccess

Puede adjuntar la política `AmazonSQSFullAccess` a las identidades de Amazon SQS. Esta política concede permisos que brindan acceso completo a Amazon SQS.

Para ver los permisos de esta política, consulte AmazonSQS FullAccess en la Referencia de políticas AWS administradas.

AWS política gestionada: AmazonSQS Access ReadOnly

Puede adjuntar la política `AmazonSQSReadOnlyAccess` a las identidades de Amazon SQS. Esta política concede permisos que brindan acceso de solo lectura a Amazon SQS.

Para ver los permisos de esta política, consulte AmazonSQS ReadOnly Access en la Referencia de políticas AWS administradas.

Amazon SQS actualiza las políticas gestionadas AWS

Consulte los detalles sobre las actualizaciones de las políticas AWS gestionadas de Amazon SQS desde que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de la API de Amazon SQS.

Cambio	Descripción	Fecha
Acceso a Amazon SQS ReadOnly	Amazon SQS ha agregado una nueva acción que permite enumerar las tareas de movimiento de mensajes más recientes (hasta diez) en una cola de origen específica. Esta acción está asociada a la operación de la API `ListMessageMoveTasks`.	9 de junio de 2023

Solución de problemas de identidad y acceso de Amazon Simple Queue Service

Utilice la siguiente información para diagnosticar y solucionar los problemas habituales que pueden surgir cuando se trabaja con Amazon SQS e IAM.

Temas

- No tengo autorización para realizar una acción en Amazon SQS
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon SQS

No tengo autorización para realizar una acción en Amazon SQS

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `sqs:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 sqs:GetWidget on resource: my-example-widget
```

En este caso, la política de Mateo se debe actualizar para permitirle acceder al recurso *`my-example-widget`* mediante la acción `sqs:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para llevar a cabo la acción `iam:PassRole`, las políticas se deben actualizar para permitirle pasar un rol a Amazon SQS.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon SQS. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon SQS

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que

asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si Amazon SQS admite estas características, consulte Cómo funciona Amazon Simple Queue Service con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro Cuenta de AWS de su propiedad en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer la diferencia entre usar roles y políticas basadas en recursos para el acceso entre cuentas, consulte el tema Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Uso de políticas con Amazon SQS

Este tema ofrece ejemplos de políticas basadas en identidad en las que un administrador de la cuenta puede adjuntar políticas de permisos a identidades de IAM (usuarios, grupos y roles).

> **Important**
>
> Le recomendamos que consulte primero los temas de introducción en los que se explican los conceptos básicos y las opciones disponibles para administrar el acceso a sus recursos de Amazon Simple Queue Service. Para obtener más información, consulte Información general sobre la administración del acceso en Amazon SQS.
>
> Con la excepción de `ListQueues`, todas las acciones de Amazon SQS admiten permisos de nivel de recurso. Para obtener más información, consulte Permisos de la API de Amazon SQS: referencia de acciones y recursos.

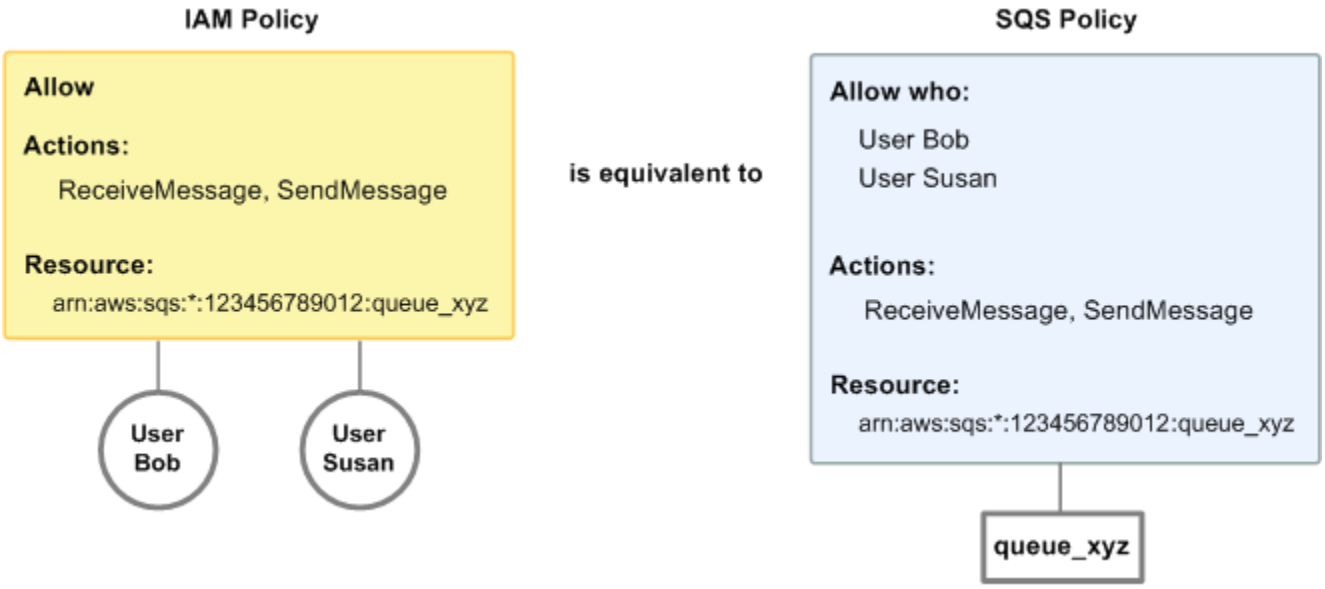
Temas

Uso de políticas de Amazon SQS e IAM

Hay dos formas de proporcionar a los usuarios permisos para los recursos de Amazon SQS: mediante el sistema de políticas de Amazon SQS y mediante el sistema de políticas de IAM. Puede aplicar uno, el otro o ambos. La mayoría de las veces, puede conseguir el mismo resultado con cualquiera de ellos.

Por ejemplo, en el siguiente diagrama se muestra una política de IAM y una política de Amazon SQS equivalente. La política de IAM otorga los derechos a Amazon `ReceiveMessage` SQS `SendMessage` y las acciones para la cola `queue_xyz` llamada en AWS su cuenta, y la política se adjunta a los usuarios llamados Bob y Susan (Bob y Susan tienen los permisos establecidos en la política). Esta política de Amazon SQS también ofrece a Bob y Susan derechos para las acciones `ReceiveMessage` y `SendMessage` de la misma cola.

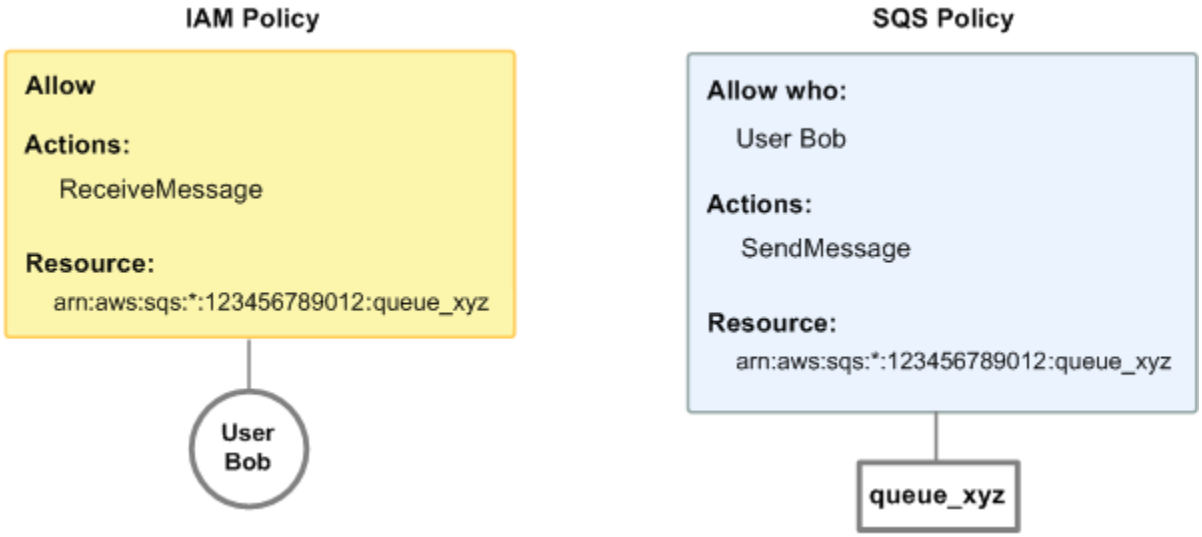
> **Note**
>
> El siguiente ejemplo muestra políticas sencillas sin condiciones. Puede especificar una condición determinada en cualquiera de las dos políticas y obtendrá el mismo resultado.



Hay una diferencia importante entre las políticas de IAM y Amazon SQS: el sistema de políticas de Amazon SQS te permite conceder permisos a AWS otras cuentas, mientras que IAM no.

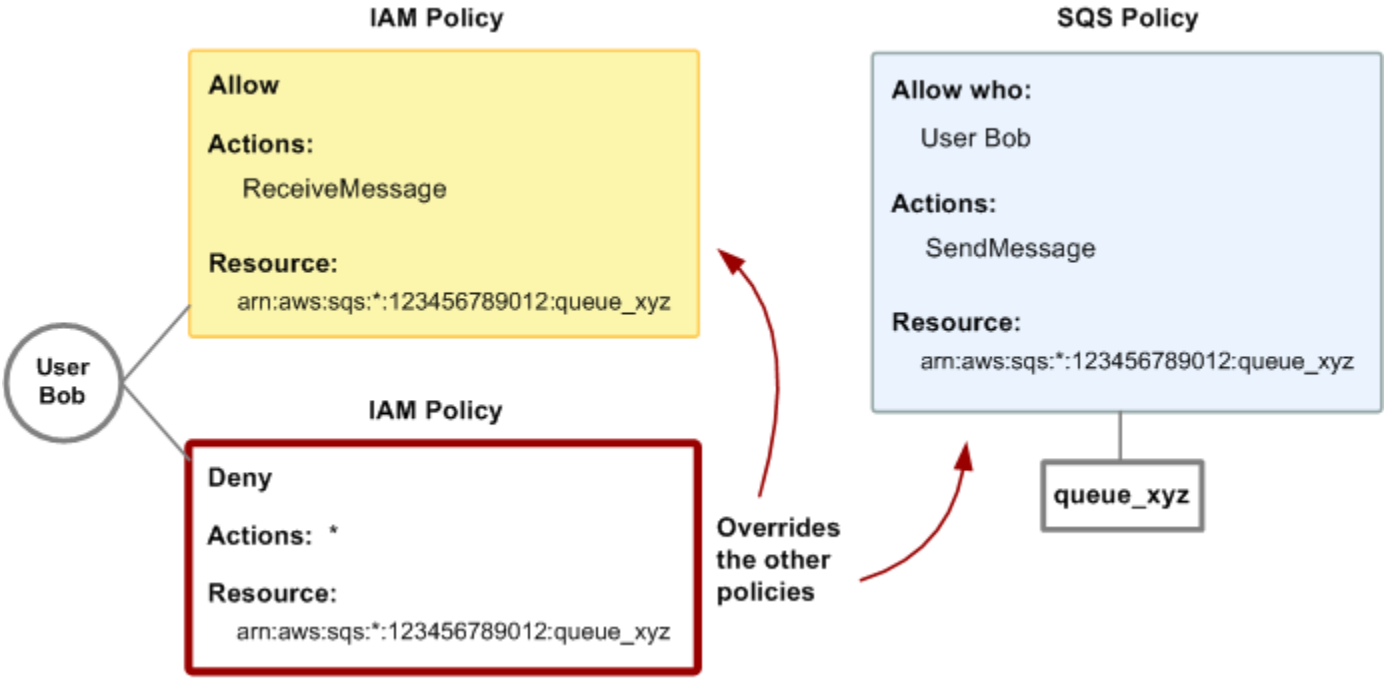
Usted decide el uso que quiere hacer de ambos sistemas para administrar los permisos. Los siguientes ejemplos muestran cómo funcionan conjuntamente los dos sistemas de política.

- En el primer ejemplo, Bob tiene una política de IAM y una política de Amazon SQS que se aplican a su cuenta. La política de IAM concede a su cuenta permiso para realizar la acción `ReceiveMessage` en `queue_xyz`, mientras que la política de Amazon SQS concede a su cuenta permiso para realizar la acción `SendMessage` en la misma cola. El siguiente diagrama ilustra este concepto.



Si Bob envía una solicitud `ReceiveMessage` a `queue_xyz`, la política de IAM permite la acción. Si Bob envía una solicitud `SendMessage` a `queue_xyz`, la política de Amazon SQS permite la acción.

- En el segundo ejemplo, Bob abusa de su acceso a `queue_xyz`, por lo que es necesario quitar todo su acceso a la cola. Para ello, lo más fácil es añadir una política que le deniegue acceso a todas las acciones de la cola. Esta política anula las otras dos porque un permiso `deny` explícito siempre anula un permiso `allow`. Para obtener más información acerca de la lógica de evaluación de las políticas, consulte Uso de políticas personalizadas con el lenguaje de la política de acceso de Amazon SQS. El siguiente diagrama ilustra este concepto.



También puede agregar a la política de Amazon SQS una instrucción adicional que deniegue a Bob cualquier tipo de acceso a la cola. Tiene el mismo efecto que agregar una política de IAM que deniegue a Bob el acceso a la cola. Para ver ejemplos de políticas que abarcan acciones y recursos de Amazon SQS, consulte Ejemplos básicos de políticas de Amazon SQS. Para obtener más información sobre la escritura de políticas de Amazon SQS, consulte Uso de políticas personalizadas con el lenguaje de la política de acceso de Amazon SQS.

Permisos necesarios para usar la consola de Amazon SQS

Un usuario que quiera trabajar con la consola de Amazon SQS debe tener el conjunto mínimo de permisos que le permita trabajar con las colas de Amazon SQS en la Cuenta de AWS del usuario.

Por ejemplo, el usuario debe tener el permiso para llamar a la acción `ListQueues` para poder enumerar colas, o el permiso para llamar a la acción `CreateQueue` para poder crear colas. Además de los permisos de Amazon SQS, para suscribir una cola de Amazon SQS a un tema de Amazon SNS, la consola también requiere permisos para las acciones de Amazon SNS.

Si crea una política de IAM que sea más restrictiva que los permisos mínimos necesarios, es posible que la consola no funcione del modo esperado para los usuarios que tienen esa política de IAM.

No es necesario permitir permisos mínimos de consola a los usuarios que solo realicen llamadas a las acciones AWS CLI o a las de Amazon SQS.

Ejemplos de políticas basadas en identidad para Amazon SQS

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon SQS. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por Amazon SQS, incluido el formato de los ARN para cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de Amazon Simple Queue Service en la Referencia de autorizaciones de servicio.

> **Note**
>
> Cuando configure los enlaces de ciclo de vida para Amazon EC2 Auto Scaling, no es necesario que escriba una política que envíe mensajes a una cola de Amazon SQS. Para obtener más información, consulte Amazon EC2 Auto Scaling Lifecycle Hooks en la Guía del usuario de Amazon EC2.

Temas

- Prácticas recomendadas sobre las políticas

- Uso de la consola de Amazon SQS
- Cómo permitir a los usuarios consultar sus propios permisos
- Permita que un usuario cree colas
- Permita a los desarrolladores escribir mensajes en una cola compartida
- Permita a los administradores obtener el tamaño general de las colas
- Permita que un socio envíe mensajes a una cola específica

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidad determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon SQS de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas

nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de Amazon SQS

Para acceder a la consola de Amazon Simple Queue Service, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de Amazon SQS que tiene en su Cuenta de AWS Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No necesita conceder permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de Amazon SQS, adjunte también la política gestionada de Amazon `AmazonSQSReadOnlyAccess` AWS SQS a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Permita que un usuario cree colas

En el siguiente ejemplo, creamos una política para Bob que le permite obtener acceso a todas las acciones de Amazon SQS, pero solo con las colas cuyos nombres tengan como prefijo la cadena literal `alice_queue_`.

Amazon SQS no concede automáticamente al creador de una cola permisos para utilizar dicha cola. Por tanto, en la política de IAM debemos conceder de forma explícita a Bob los permisos para utilizar todas las acciones de Amazon SQS, además de para la acción `CreateQueue`.

```
{
```

```
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:*:123456789012:alice_queue_*"
   }]
}
```

Permita a los desarrolladores escribir mensajes en una cola compartida

En el siguiente ejemplo, creamos un grupo para desarrolladores y adjuntamos una política que permite al grupo usar la `SendMessage` acción Amazon SQS, pero solo con la cola que pertenece a la especificada Cuenta de AWS y que lleva el nombre. `MyCompanyQueue`

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:*:123456789012:MyCompanyQueue"
   }]
}
```

Puede utilizar * en lugar de `SendMessage` para conceder las acciones siguientes de una cola compartida a una entidad principal: `ChangeMessageVisibility`, `DeleteMessage`, `GetQueueAttributes`, `GetQueueUrl`, `ReceiveMessage` y `SendMessage`.

> **Note**
>
> Aunque * incluye el acceso proporcionado por otros tipos de permisos, Amazon SQS considera los permisos por separado. Por ejemplo, es posible conceder los permisos * y `SendMessage` a un usuario, aunque * incluye el acceso que ofrece `SendMessage`. Este concepto también se aplica al quitar un permiso. Si una entidad principal solo tiene el permiso * y se solicita la eliminación del permiso `SendMessage`, la entidad principal no se queda con los permisos restantes, sino que la solicitud no tiene ningún efecto, porque la entidad principal no tenía explícitamente el permiso `SendMessage`. Si desea dejar a la entidad principal únicamente con el permiso `ReceiveMessage`, añada primero el permiso `ReceiveMessage` y, a continuación, elimine el permiso *.

Permita a los administradores obtener el tamaño general de las colas

En el siguiente ejemplo, creamos un grupo para administradores y adjuntamos una política que permite al grupo usar la `GetQueueAttributes` acción Amazon SQS con todas las colas que pertenecen a la cuenta especificada. AWS

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "sqs:GetQueueAttributes",
      "Resource": "*"
   }]
}
```

Permita que un socio envíe mensajes a una cola específica

Puede realizar esta tarea mediante una política de Amazon SQS o una política de IAM. Si su socio tiene una Cuenta de AWS, podría ser más fácil usar una política de Amazon SQS. Sin embargo, cualquier usuario de la empresa del socio que posea las credenciales AWS de seguridad puede enviar mensajes a la cola. Si desea limitar el acceso a un determinado usuario o aplicación, debe tratar al socio como a un usuario de su propia empresa y utilizar una política de IAM en lugar de una política de Amazon SQS.

En este ejemplo se realizan las siguientes acciones:

1. Cree un grupo llamado WidgetCo para representar a la empresa asociada.
2. Crear un usuario para la aplicación o el usuario específico de la compañía del socio que necesita acceso.
3. Agregue el usuario al grupo .
4. Adjuntar una política que conceda al grupo acceso únicamente a la acción `SendMessage` solo para la cola denominada `WidgetPartnerQueue`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
         "Effect": "Allow",
         "Action": "sqs:SendMessage",
```

```
         "Resource": "arn:aws:sqs:*:123456789012:WidgetPartnerQueue"
   }]
}
```

Ejemplos básicos de políticas de Amazon SQS

En esta sección se muestran políticas de ejemplo para casos de uso Amazon SQS comunes.

Puede utilizar la consola para comprobar los efectos de cada política a medida que asocia la política al usuario. En un primer momento, como el usuario no tiene permisos, no podrá hacer nada más en la consola. Al asignar políticas al usuario, podrá verificar que este pueda realizar diversas acciones en la consola.

> **Note**
>
> Le recomendamos que utilice dos ventanas del navegador: una para conceder los permisos y otra para iniciar sesión AWS Management Console con las credenciales del usuario a fin de comprobar los permisos a medida que se los concede al usuario.

Ejemplo 1: conceder un permiso a otro Cuenta de AWS

El siguiente ejemplo de política otorga a Cuenta de AWS `111122223333` Number el `SendMessage` permiso para la cola nombrada `444455556666/queue1` en la región EE.UU. Este (Ohio).

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_SendMessage",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "111122223333"
         ]
      },
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:us-east-2:444455556666:queue1"
   }]
}
```

Ejemplo 2: conceder dos permisos a uno Cuenta de AWS

El siguiente ejemplo de política concede `111122223333` al Cuenta de AWS número `SendMessage` y el `ReceiveMessage` permiso para la cola denominada`444455556666/queue1`.

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_Send_Receive",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "111122223333"
         ]
      },
      "Action": [
         "sqs:SendMessage",
         "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:*:444455556666:queue1"
   }]
}
```

Ejemplo 3: conceder todos los permisos a dos Cuentas de AWS

El siguiente ejemplo de política concede dos Cuentas de AWS números diferentes (`111122223333`y`444455556666`) permiso para utilizar todas las acciones a las que Amazon SQS permite el acceso compartido para la cola nombrada `123456789012/queue1` en la región EE.UU. Este (Ohio).

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_AllActions",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "111122223333",
            "444455556666"
         ]
```

```
      },
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:us-east-2:123456789012:queue1"
   }]
}
```

Ejemplo 4: conceder permisos entre cuentas a un rol y un nombre de usuario

El siguiente ejemplo de política concede `role1` un permiso `111122223333` multicuenta `username1` bajo Cuenta de AWS número para utilizar todas las acciones a las que Amazon SQS permite el acceso compartido para la cola `123456789012/queue1` nombrada en la región EE.UU. Este (Ohio).

Los permisos entre cuentas no se aplican a las siguientes acciones:

- `AddPermission`
- `CancelMessageMoveTask`
- `CreateQueue`
- `DeleteQueue`
- `ListMessageMoveTask`
- `ListQueues`
- `ListQueueTags`
- `RemovePermission`
- `SetQueueAttributes`
- `StartMessageMoveTask`
- `TagQueue`
- `UntagQueue`

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_AllActions",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "arn:aws:iam::111122223333:role/role1",
```

```
            "arn:aws:iam::111122223333:user/username1"
         ]
      },
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:us-east-2:123456789012:queue1"
   }]
}
```

Ejemplo 5: conceder un permiso a todos los usuarios

La siguiente política de ejemplo concede a todos los usuarios (usuarios anónimos) el permiso `ReceiveMessage` para la cola denominada `111122223333/queue1`.

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_AnonymousAccess_ReceiveMessage",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:ReceiveMessage",
      "Resource": "arn:aws:sqs:*:111122223333:queue1"
   }]
}
```

Ejemplo 6: conceder un permiso con restricción temporal a todos los usuarios

La siguiente política de ejemplo concede a todos los usuarios (usuarios anónimos) el permiso `ReceiveMessage` para la cola denominada `111122223333/queue1`, pero solo desde las 12:00 h (mediodía) hasta las 15:00 h el 31 de enero de 2009.

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_AnonymousAccess_ReceiveMessage_TimeLimit",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:ReceiveMessage",
      "Resource": "arn:aws:sqs:*:111122223333:queue1",
      "Condition" : {
         "DateGreaterThan" : {
```

```
            "aws:CurrentTime":"2009-01-31T12:00Z"
         },
         "DateLessThan" : {
            "aws:CurrentTime":"2009-01-31T15:00Z"
         }
      }
   }]
}
```

Ejemplo 7: conceder todos los permisos a todos los usuarios de un rango de CIDR

La siguiente política de ejemplo concede a todos los usuarios (usuarios anónimos) permiso para utilizar todas las acciones posibles de Amazon SQS que se pueden compartir para la cola denominada `111122223333/queue1`, pero solo si la solicitud procede del intervalo de CIDR `192.0.2.0/24`.

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_AnonymousAccess_AllActions_AllowlistIP",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:*:111122223333:queue1",
      "Condition" : {
         "IpAddress" : {
            "aws:SourceIp":"192.0.2.0/24"
         }
      }
   }]
}
```

Ejemplo 8: permisos Allowlist y blocklist para los usuarios de diferentes rangos de CIDR

La siguiente política de ejemplo contiene dos instrucciones:

- La primera instrucción concede a todos los usuarios (usuarios anónimos) del rango de CIDR `192.0.2.0/24` (excepto `192.0.2.188`) permiso para utilizar la acción `SendMessage` para la cola denominada `111122223333`/queue1.
- La segunda instrucción bloquea a todos los usuarios (usuarios anónimos) del rango de CIDR `12.148.72.0/23` y les impide utilizar la cola.

```
{
   "Version": "2012-10-17",
   "Id": "Queue1_Policy_UUID",
   "Statement": [{
      "Sid":"Queue1_AnonymousAccess_SendMessage_IPLimit",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:*:111122223333:queue1",
      "Condition" : {
         "IpAddress" : {
            "aws:SourceIp":"192.0.2.0/24"
         },
         "NotIpAddress" : {
            "aws:SourceIp":"192.0.2.188/32"
         }
      }
   }, {
      "Sid":"Queue1_AnonymousAccess_AllActions_IPLimit_Deny",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:*:111122223333:queue1",
      "Condition" : {
         "IpAddress" : {
            "aws:SourceIp":"12.148.72.0/23"
         }
      }
   }]
}
```

Uso de políticas personalizadas con el lenguaje de la política de acceso de Amazon SQS

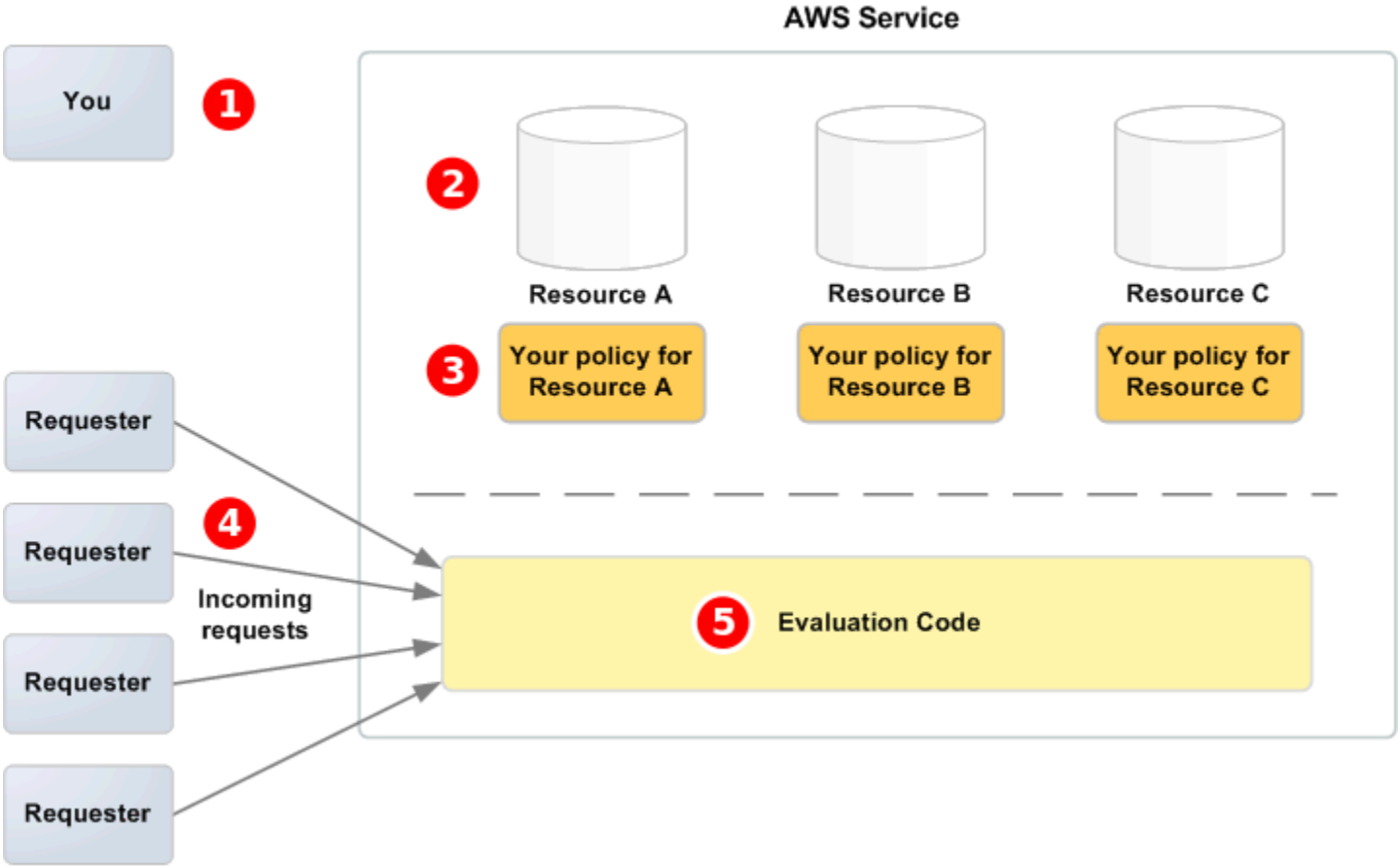
Si quiere permitir el acceso a Amazon SQS basándose únicamente en un Cuenta de AWS ID y permisos básicos (por ejemplo, para `SendMessage`o `ReceiveMessage`), no necesita escribir sus propias políticas. Basta con que utilice la acción `AddPermission` de Amazon SQS.

Si quiere denegar o permitir el acceso de forma explícita en función de condiciones más específicas (como la hora en que llega la solicitud o la dirección IP del solicitante), debe redactar sus propias políticas de Amazon SQS y cargarlas en AWS el sistema mediante la acción Amazon SQS. `SetQueueAttributes`

Temas

Arquitectura de control de acceso de Amazon SQS

En el diagrama siguiente, se describe el control de acceso para los recursos de Amazon SQS.



1

Usted, el propietario del recurso.

2 Sus recursos contenidos en el AWS servicio (por ejemplo, las colas de Amazon SQS).

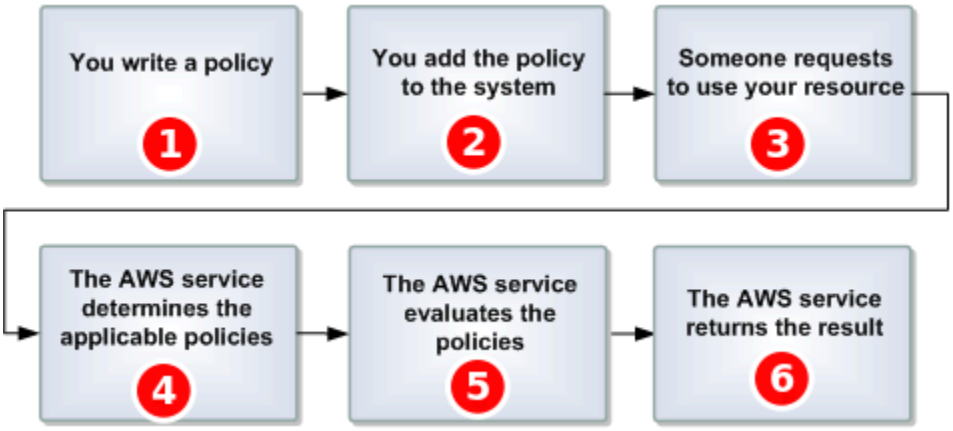
3 Sus políticas. Es recomendable tener una política por recurso. El AWS servicio proporciona una API que puede utilizar para cargar y gestionar sus políticas.

4 Los solicitantes y las solicitudes de entrada dirigidas al servicio de AWS .

5 El código de evaluación del lenguaje de la política de acceso. Se trata del conjunto de códigos del AWS servicio que evalúa las solicitudes entrantes comparándolas con las políticas aplicables y determina si el solicitante puede acceder al recurso.

Flujo de trabajo del proceso de control de acceso de Amazon SQS

En el diagrama siguiente, se describe el flujo de trabajo general del control de acceso con el lenguaje de la política de acceso de Amazon SQS.



1 Escribe una política de Amazon SQS para la cola.

2 Debe cargar su política en. AWS El AWS servicio proporciona una API que puedes usar para cargar tus políticas. Por ejemplo, utilice la acción `SetQueueAttributes` de Amazon SQS a fin de cargar una política para una cola de Amazon SQS determinada.

3

Alguien envía una solicitud para utilizar su cola de Amazon SQS.

4

Amazon SQS examina todas las políticas de Amazon SQS disponibles y determina cuáles son aplicables.

5

Amazon SQS evalúa las políticas y determina si el solicitante tiene permiso para utilizar su cola.

6

En función del resultado de evaluar las políticas, Amazon SQS devuelve un error `Access denied` al solicitante o continúa procesando la solicitud.

Conceptos clave del lenguaje de la política de acceso de Amazon SQS

Para escribir sus propias políticas, debe estar familiarizado con JSON y varios conceptos clave.

Permitir

El resultado de una Instrucción cuyo Effect se ha establecido en `allow`.

Action

La actividad que la Entidad principal tiene permiso para realizar, por lo general una solicitud para AWS.

Default-deny

Resultado de una Instrucción que no tiene la opción Permitir o Explicit-deny.

Condición

Cualquier restricción o detalle sobre un Permiso. Las condiciones típicas están relacionados con la fecha y la hora y con las direcciones IP.

Effect

El resultado que desea que la Instrucción de una Política devuelva cuando se evalúe. Al escribir la instrucción de la política, debe especificar el valor `deny` o `allow`. Puede haber tres resultados posibles cuando se evalúe la política: Default-deny, Permitir y Explicit-deny.

Explicit-deny

El resultado de una Instrucción cuyo Effect se ha establecido en `deny`.

Evaluación

El proceso que Amazon SQS utiliza para determinar si una solicitud de entrada se debe denegar o permitir en función de una Política.

Emisor

El usuario que escribe una Política para conceder permisos para un recurso. El emisor, por definición, es siempre el propietario del recurso. AWS no permite a los usuarios de Amazon SQS crear políticas para recursos que no son de su propiedad.

Clave

La característica específica que es la base para la restricción del acceso.

Permiso

El concepto de permitir o impedir el acceso a un recurso mediante una Condición y una Clave.

Política

El documento que actúa como contenedor de una o varias instrucciones.

Amazon SQS utiliza la política para determinar si desea conceder acceso a un usuario para un recurso.

Entidad principal

El usuario que recibe el Permiso en la Política.

Resource

El objeto para el que la Entidad principal solicita acceso.

Instrucción

Descripción formal de un único permiso, escrita en el lenguaje de la política de acceso como parte de un documento más amplio de la Política.

Solicitante

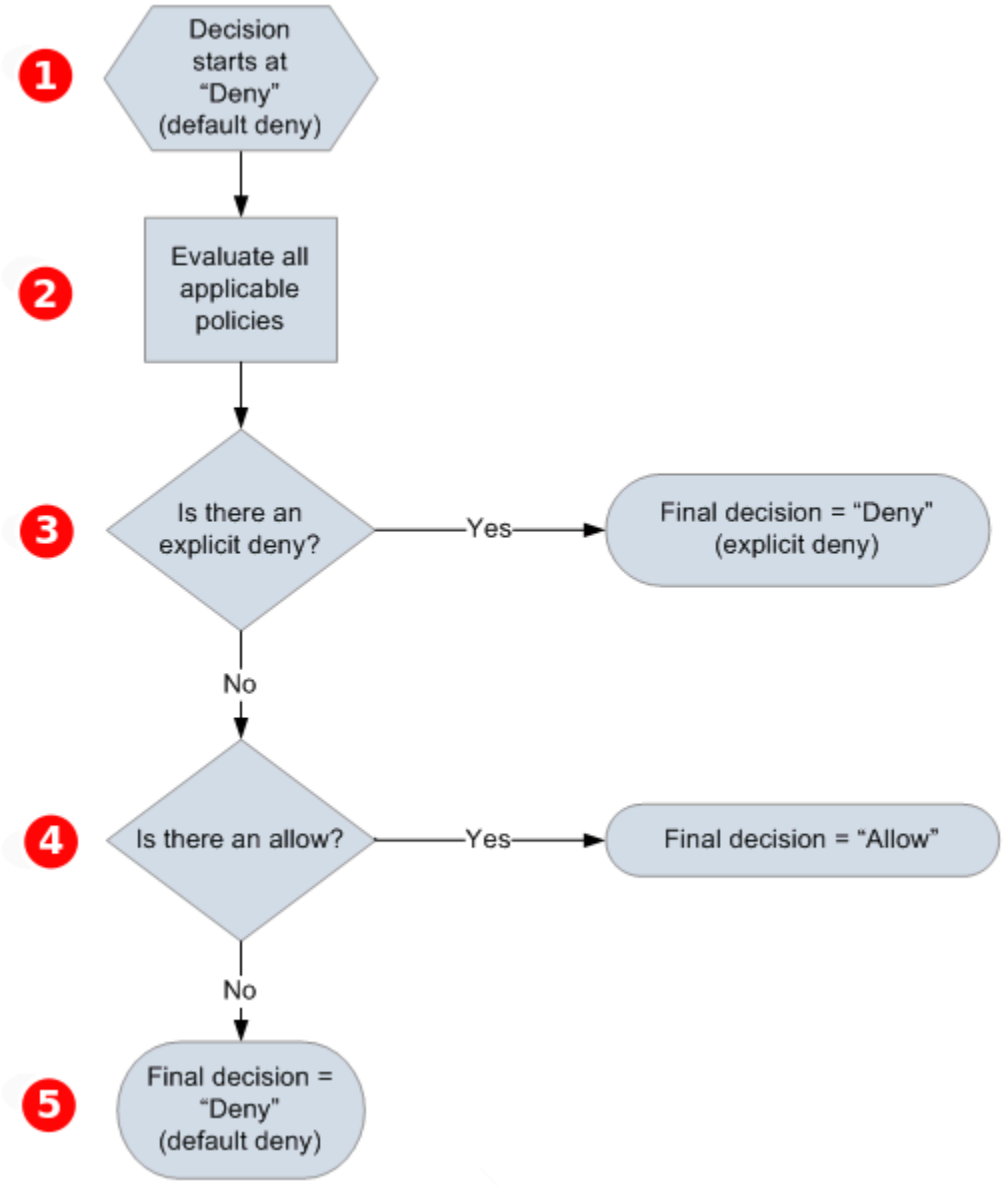
El usuario que envía una solicitud de acceso a un Resource.

Lógica de evaluación del lenguaje de la política de acceso de Amazon SQS

En el momento de la evaluación, Amazon SQS determina si las solicitudes procedentes de una persona que no es la propietaria del recurso deben permitirse o denegarse. La lógica de evaluación sigue varias reglas básicas:

- De forma predeterminada, todas las solicitudes para utilizar su recurso procedentes de cualquier persona que no sea usted se deniegan.
- Un valor Permitir anula cualquier valor Default-deny.
- Una instrucción Explicit-deny anula cualquier instrucción allow.
- El orden en que se evalúan las políticas no es importante.

En el diagrama siguiente, se describe con detalle cómo Amazon SQS evalúa las decisiones sobre los permisos de acceso.



1

La decisión comienza con una instrucción default-deny.

2

El código de aplicación evalúa todas las políticas aplicables a la solicitud (en función del recurso, la entidad principal, la acción y las condiciones). El orden en que el código de aplicación evalúa las políticas no es importante.

3

El código de aplicación busca una instrucción explicit-deny que se puede aplicar a la solicitud. Si encuentra alguna, devuelve una decisión de denegar y el proceso termina.

4

Si no se encuentra ninguna instrucción explicit-deny, el código de cumplimiento buscará una instrucción allow que se pueda aplicar a la solicitud. Si encuentra alguna, el código de aplicación devuelve una decisión de permitir y el proceso continúa (el servicio sigue procesando la solicitud).

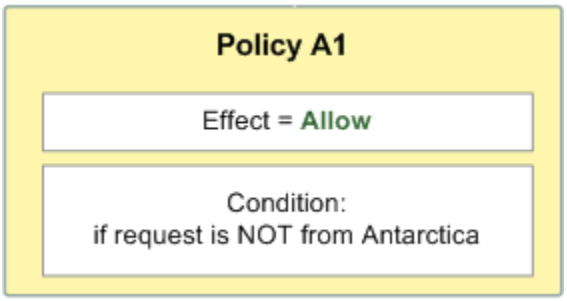
5

Si no se encuentra ninguna instrucción allow, la decisión final será deny (como no hay ninguna instrucción explicit-deny o allow, se considera una instrucción default-deny).

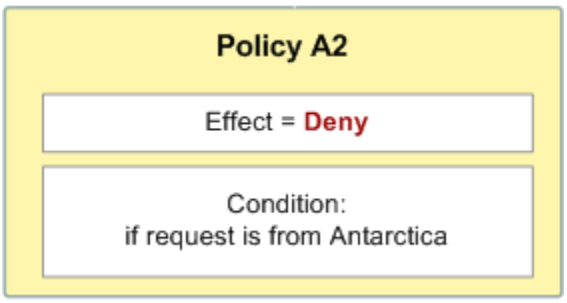
Relaciones entre denegaciones explícitas y predeterminadas en el lenguaje de la política de acceso de Amazon SQS

Si una política de Amazon SQS no se aplica directamente a una solicitud, esta da como resultado Default-deny. Por ejemplo, si un usuario solicita permiso para utilizar Amazon SQS pero la única política que se aplica al usuario puede utilizar DynamoDB, las solicitudes dan como resultado default-deny.

Si no se cumple una condición de una instrucción, la solicitud da como resultado default-deny. Si se cumplen todas las condiciones de una instrucción, la solicitud da como resultado Permitir o Explicit-deny, en función del valor del elemento Effect de la política. Las políticas no especifican qué es lo que se debe hacer si no se cumple una condición, por lo que el resultado predeterminado en este caso es default-deny. Por ejemplo, suponga que desea evitar las solicitudes que proceden de la Antártida. Puede escribir una política denominada A1 que permite una solicitud solo si no procede de la Antártida. El siguiente diagrama ilustra la política de Amazon SQS.

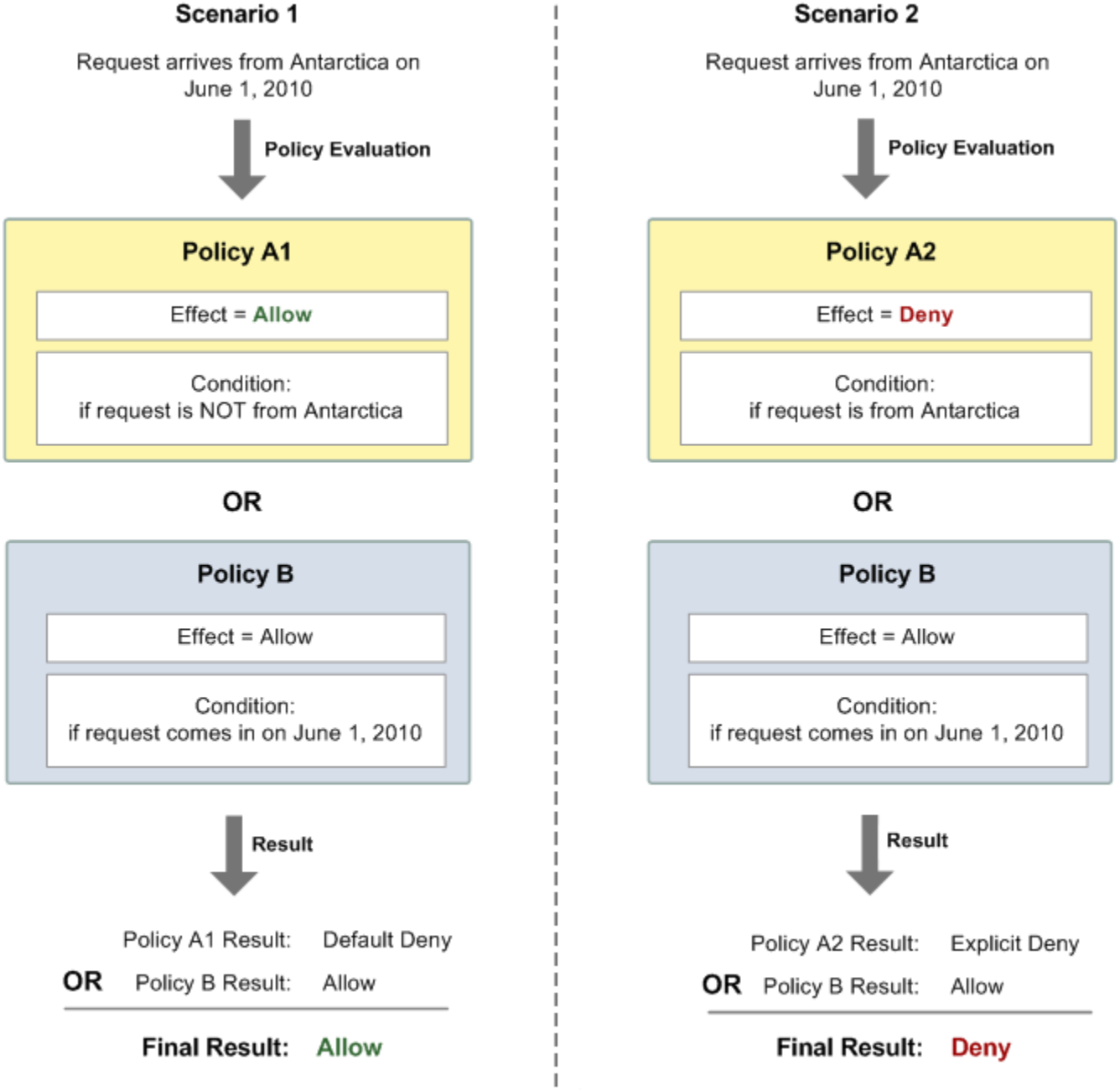


Si un usuario envía una solicitud desde Estados Unidos, la condición se cumple (la solicitud no procede de la Antártida) y la solicitud da como resultado allow. Sin embargo, si un usuario envía una solicitud desde la Antártida, la condición no se cumple y la solicitud da como resultado de forma predeterminada default-deny. Puede cambiar el resultado a explicit-deny si escribe una política A2 que deniegue explícitamente una solicitud si procede de la Antártida. El siguiente diagrama ilustra esta política.



Si un usuario envía una solicitud desde la Antártida, la condición se cumple y la solicitud da como resultado explicit-deny.

La diferencia entre default-deny y explicit-deny es importante, ya que allow puede sobrescribir la primera, pero no la segunda. Por ejemplo, la política B permite solicitudes si llegan el 1 de junio de 2010. En el siguiente diagrama se compara la combinación de esta política con las políticas A1 y A2.



En el escenario 1, la política A1 da lugar a default-deny y la política B da lugar a allow, ya que la política permite las solicitudes que entran el 1 de junio de 2010. La instrucción allow de la política B anula la instrucción default-deny de la política A1 y la solicitud se permite.

En el escenario 2, la política B2 da lugar a explicit-deny y la política B da lugar a allow. La instrucción explicit-deny de la política A2 anula la instrucción allow de la política B y la solicitud se deniega.

Limitaciones de las políticas personalizadas de Amazon SQS

Acceso entre cuentas

Los permisos entre cuentas no se aplican a las siguientes acciones:

- `AddPermission`
- `CancelMessageMoveTask`
- `CreateQueue`
- `DeleteQueue`
- `ListMessageMoveTask`
- `ListQueues`
- `ListQueueTags`
- `RemovePermission`
- `SetQueueAttributes`
- `StartMessageMoveTask`
- `TagQueue`
- `UntagQueue`

Claves de condición

En la actualidad, Amazon SQS únicamente admite un subconjunto limitado de claves de condición disponibles en IAM. Para obtener más información, consulte Permisos de la API de Amazon SQS: referencia de acciones y recursos.

Ejemplos de lenguaje de la política de acceso de Amazon SQS personalizada

A continuación, se muestran algunos ejemplos de políticas de acceso de Amazon SQS típicas.

Ejemplo 1: conceder permiso a una cuenta

En el ejemplo siguiente, la política de Amazon SQS proporciona a la `queue2` 111122223333 permiso para enviar y recibir información de la cola Cuenta de AWS , que es propiedad de la Cuenta de AWS 444455556666.

```
{
   "Version": "2012-10-17",
```

```
   "Id": "UseCase1",
   "Statement" : [{
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "111122223333"
         ]
      },
      "Action": [
         "sqs:SendMessage",
         "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-2:444455556666:queue2"
   }]
}
```

Ejemplo 2: conceder permiso a una o varias cuentas

El siguiente ejemplo de política de Amazon SQS proporciona uno o más Cuentas de AWS accesos a las colas que son propiedad de su cuenta durante un período de tiempo específico. Es necesario escribir esta política y cargarla en Amazon SQS mediante la acción `SetQueueAttributes`, ya que la acción `AddPermission` no permite especificar una restricción de tiempo al conceder acceso a una cola.

```
{
   "Version": "2012-10-17",
   "Id": "UseCase2",
   "Statement" : [{
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "111122223333",
            "444455556666"
         ]
      },
      "Action": [
         "sqs:SendMessage",
         "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-2:444455556666:queue2",
      "Condition": {
```

```
         "DateLessThan": {
            "AWS:CurrentTime": "2009-06-30T12:00Z"
         }
      }
   }]
}
```

Ejemplo 3: conceder permiso a solicitudes de instancias de Amazon EC2

La siguiente política de ejemplo de Amazon SQS concede acceso a las solicitudes que proceden de instancias de Amazon EC2. Este ejemplo se basa en el ejemplo "Ejemplo 2: conceder permiso a una o varias cuentas": restringe el acceso antes del 30 de junio de 2009 a las 12:00 h (UTC) y al rango de direcciones IP `203.0.113.0/24`. Es necesario escribir esta política y cargarla en Amazon SQS mediante la acción `SetQueueAttributes`, porque la acción `AddPermission` no permite especificar una restricción de dirección IP al conceder acceso a una cola.

```
{
   "Version": "2012-10-17",
   "Id": "UseCase3",
   "Statement" : [{
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "111122223333"
         ]
      },
      "Action": [
         "sqs:SendMessage",
         "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-2:444455556666:queue2",
      "Condition": {
         "DateLessThan": {
            "AWS:CurrentTime": "2009-06-30T12:00Z"
         },
         "IpAddress": {
            "AWS:SourceIp": "203.0.113.0/24"
         }
      }
   }]
}
```

Ejemplo 4: denegar acceso a una cuenta específica

El siguiente ejemplo de política de Amazon SQS deniega un Cuenta de AWS acceso específico a su cola. Este ejemplo se basa en el ejemplo «Ejemplo 1: conceder permiso a una cuenta»: deniega el acceso a lo especificado. Cuenta de AWS Es necesario escribir esta política y cargarla en Amazon SQS mediante la acción `SetQueueAttributes`, ya que la acción `AddPermission` no permite denegar el acceso a una cola (solo permite conceder acceso a una cola).

```
{
   "Version": "2012-10-17",
   "Id": "UseCase4",
   "Statement" : [{
      "Sid": "1",
      "Effect": "Deny",
      "Principal": {
         "AWS": [
            "111122223333"
         ]
      },
      "Action": [
         "sqs:SendMessage",
         "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-2:444455556666:queue2"
   }]
}
```

Ejemplo 5: denegar el acceso si no es desde un punto de enlace de la VPC

La siguiente política de ejemplo de Amazon SQS restringe el acceso a `queue1`: 111122223333 puede realizar las acciones `SendMessage` y `ReceiveMessage` solo desde el ID `vpce-1a2b3c4d` del punto de conexión de VPC (especificado mediante la condición `aws:sourceVpce`). Para obtener más información, consulte Puntos de conexión de Amazon Virtual Private Cloud para Amazon SQS.

> **Note**
>
> - La condición `aws:sourceVpce` no requiere un ARN para el recurso de punto de enlace de la VPC, solo el ID de la VPC.
> - Puede modificar el siguiente ejemplo para restringir todas las acciones para un punto de conexión de VPC concreto mediante la denegación de todas las acciones de Amazon SQS

(`sqs:*`) en la segunda instrucción. Sin embargo, una declaración de política de este tipo estipularía que todas las acciones (incluidas las acciones administrativas necesarias para modificar los permisos de cola) deben realizarse a través del punto de enlace de la VPC específico definido en la política, lo que podría impedir al usuario de la cola modificar los permisos en el futuro.

```
{
   "Version": "2012-10-17",
   "Id": "UseCase5",
   "Statement": [{
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
         "AWS": [
            "111122223333"
         ]
      },
      "Action": [
         "sqs:SendMessage",
         "sqs:ReceiveMessage"
      ],
         "Resource": "arn:aws:sqs:us-east-2:111122223333:queue1"
      },
      {
         "Sid": "2",
         "Effect": "Deny",
         "Principal": "*",
         "Action": [
            "sqs:SendMessage",
            "sqs:ReceiveMessage"
         ],
         "Resource": "arn:aws:sqs:us-east-2:111122223333:queue1",
         "Condition": {
            "StringNotEquals": {
               "aws:sourceVpce": "vpce-1a2b3c4d"
            }
         }
      }
   ]
}
```

Uso de credenciales de seguridad temporales con Amazon SQS

Además de crear usuarios con sus propias credenciales de seguridad, IAM también le permite conceder credenciales de seguridad temporales a cualquier usuario, lo que le permite acceder a sus AWS servicios y recursos. Puede administrar usuarios que tengan Cuentas de AWS. También puede administrar los usuarios de su sistema que no los tengan Cuentas de AWS (usuarios federados). Además, las aplicaciones que cree para acceder a sus AWS recursos también se pueden considerar «usuarios».

Puede utilizar estas credenciales de seguridad temporales para realizar solicitudes a Amazon SQS. Las bibliotecas de la API computan el valor de firma necesario con esas credenciales para autenticar su solicitud. Si envía las solicitudes con las credenciales caducadas, Amazon SQS deniega la solicitud.

> **Note**
>
> No se puede establecer una política basada en credenciales temporales.

Requisitos previos

1. Utilice IAM para crear credenciales de seguridad temporales:
   - Token de seguridad
   - ID de clave de acceso)
   - Clave de acceso secreta
2. Prepare la cadena para firmar con el ID de clave de acceso temporal y el token de seguridad.
3. Utilice la clave de acceso secreta temporal en lugar de su propia clave de acceso secreta para firmar la solicitud de la API de consultas.

> **Note**
>
> Al enviar la solicitud de la API de consultas firmada, utilice el ID de clave de acceso temporal en lugar de su propio ID de clave de acceso e incluya el token de seguridad. Para obtener más información sobre la compatibilidad de IAM con las credenciales de seguridad temporales, consulte Cómo conceder acceso temporal a sus AWS recursos en la Guía del usuario de IAM.

Para llamar a una acción de la API de consultas de Amazon SQS mediante credenciales de seguridad temporales

1. Solicite un token de seguridad temporal utilizando. AWS Identity and Access Management Para obtener más información, consulte Creación de credenciales de seguridad temporales para permitir el acceso a los usuarios de IAM en la Guía del usuario de IAM.

   IAM devuelve un token de seguridad, un ID de clave de acceso y una clave de acceso secreta.
2. Prepare la consulta con el ID de clave de acceso temporal en lugar de su propio ID de clave de acceso e incluya el token de seguridad. Firme la solicitud con la clave de acceso secreta temporal en lugar de utilizar su propia clave.
3. Envíe la cadena de consulta firmada con el ID de clave de acceso temporal y el token de seguridad.

   En el siguiente ejemplo, se muestra cómo se utilizan las credenciales de seguridad temporales para autenticar una solicitud de Amazon SQS. La estructura de *AUTHPARAMS* depende de la firma de la solicitud de API. Para obtener más información, consulte Firmar solicitudes de AWS API en la Referencia general de Amazon Web Services.

   ```
   https://sqs.us-east-2.amazonaws.com/
   ?Action=CreateQueue
   &DefaultVisibilityTimeout=40
   &QueueName=MyQueue
   &Attribute.1.Name=VisibilityTimeout
   &Attribute.1.Value=40
   &Expires=2020-12-18T22%3A52%3A43PST
   &SecurityToken=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   &AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
   &Version=2012-11-05
   &AUTHPARAMS
   ```

   El siguiente ejemplo utiliza credenciales de seguridad temporales para enviar dos mensajes con la acción `SendMessageBatch`.

   ```
   https://sqs.us-east-2.amazonaws.com/
   ?Action=SendMessageBatch
   &SendMessageBatchRequestEntry.1.Id=test_msg_001
   &SendMessageBatchRequestEntry.1.MessageBody=test%20message%20body%201
   &SendMessageBatchRequestEntry.2.Id=test_msg_002
   &SendMessageBatchRequestEntry.2.MessageBody=test%20message%20body%202
   ```

```
&SendMessageBatchRequestEntry.2.DelaySeconds=60
&Expires=2020-12-18T22%3A52%3A43PST
&SecurityToken=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
&AWSAccessKeyId=AKIAI44QH8DHBEXAMPLE
&Version=2012-11-05
&AUTHPARAMS
```

Gestión de acceso para colas cifradas de Amazon SQS con políticas de privilegios mínimos

Puede utilizar Amazon SQS para intercambiar datos confidenciales entre aplicaciones mediante el cifrado del servidor (SSE) integrado con AWS Key Management Service (KMS). Con la integración de Amazon SQS AWS KMS, puede gestionar de forma centralizada las claves que protegen Amazon SQS, así como las claves que protegen sus demás recursos. AWS

Varios AWS servicios pueden actuar como fuentes de eventos que envían eventos a Amazon SQS. Para permitir que una fuente de eventos acceda a la cola cifrada de Amazon SQS, debe configurar la cola con una clave gestionada por el cliente. AWS KMS A continuación, utilice la política de claves para permitir que el servicio utilice los métodos de API necesarios. AWS KMS El servicio también requiere permisos de autenticación de acceso para que la cola pueda enviar eventos. Puede conseguirlo mediante una política de Amazon SQS, que es una política basada en recursos que puede utilizar para controlar el acceso a la cola de Amazon SQS y a sus datos.

En las siguientes secciones se proporciona información sobre cómo controlar el acceso a la cola cifrada de Amazon SQS mediante la política de Amazon SQS y la política de claves. AWS KMS Las políticas de esta guía le ayudarán a conseguir el privilegio mínimo.

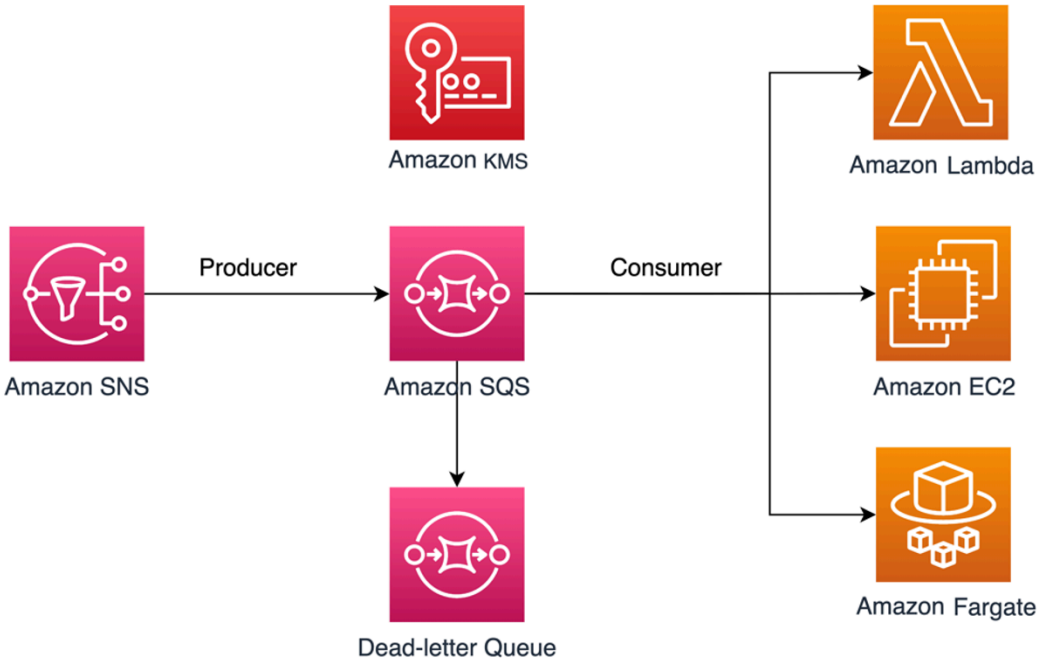
En esta guía también se describe cómo las políticas basadas en recursos abordan el problema del suplente confuso mediante la utilización de las claves de contexto de condición de IAM globales `aws:SourceArn`, `aws:SourceAccount` y `aws:PrincipalOrgID`.

Temas

- Información general
- Política de claves de privilegio mínimo para Amazon SQS
- Instrucciones de política de Amazon SQS para la cola de mensajes fallidos
- Prevención del problema del suplente confuso entre servicios
- Uso de Analizador de acceso de IAM para revisar el acceso entre cuentas

Información general

En este tema, le guiaremos a través de un caso de uso común para ilustrar cómo puede crear la política de claves y la política de colas de Amazon SQS. Este caso de uso se muestra en la siguiente imagen.



En este ejemplo, el productor de mensajes es un tema de Amazon Simple Notification Service (SNS) que está configurado para la distribución ramificada de mensajes a su cola cifrada de Amazon SQS. El consumidor de mensajes es un servicio de computación, como una función de AWS Lambda, una instancia de Amazon Elastic Compute Cloud (EC2) o un contenedor de AWS Fargate. A continuación, la cola de Amazon SQS se configura para enviar los mensajes fallidos a una cola de mensajes fallidos (DLQ). Esto es útil para depurar su aplicación o sistema de mensajería porque las DLQ le permiten aislar los mensajes no consumidos para determinar por qué su procesamiento no se realizó correctamente. En la solución definida en este tema, se utiliza un servicio de computación como una función de Lambda para procesar los mensajes almacenados en la cola de Amazon SQS. Si el consumidor de mensajes se encuentra en una nube privada virtual (VPC), la instrucción de política `DenyReceivingIfNotThroughVPCE` incluida en esta guía le permite restringir la recepción de mensajes a esa VPC específica.

> **Note**
>
> Esta guía contiene solo los permisos de IAM necesarios en forma de instrucciones de política. Para crear la política, debe añadir las declaraciones a su política de Amazon SQS

o a su política AWS KMS clave. Esta guía no proporciona instrucciones sobre cómo crear la cola o la clave de Amazon SQS. AWS KMS Para obtener instrucciones sobre cómo crear estos recursos, consulte Creación de una cola de Amazon SQS y Creación de claves. La política de Amazon SQS definida en esta guía no admite el redireccionamiento de mensajes directamente a la misma cola de Amazon SQS o a una cola diferente.

Política de claves de privilegio mínimo para Amazon SQS

En esta sección, describimos los permisos con privilegios mínimos necesarios AWS KMS para la clave administrada por el cliente que utiliza para cifrar la cola de Amazon SQS. Con estos permisos, puede limitar el acceso solo a las entidades previstas a la vez que implementa el privilegio mínimo. La política de claves debe constar de las siguientes instrucciones de política, que describimos detalladamente a continuación:

- Otorgue permisos de administrador a la clave AWS KMS
- Concesión de acceso de solo lectura a los metadatos de clave
- Concesión de permisos de KMS de Amazon SNS a Amazon SNS para publicar mensajes en la cola
- Permiso a los consumidores para descifrar mensajes de la cola

Otorgue permisos de administrador a la clave AWS KMS

Para crear una AWS KMS clave, debe proporcionar permisos de AWS KMS administrador a la función de IAM que utilice para implementar la AWS KMS clave. Estos permisos de administrador se definen en la siguiente instrucción de política de `AllowKeyAdminPermissions`. Cuando añadas esta declaración a tu política de AWS KMS claves, asegúrate de <admin-role ARN>sustituirla por el nombre de recurso de Amazon (ARN) de la función de IAM utilizada para implementar la AWS KMS clave, gestionar la AWS KMS clave o ambas. Puede ser el rol de IAM de su canalización de implementación o el rol de administrador de su organización en AWS Organizations.

```
{
  "Sid": "AllowKeyAdminPermissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "<admin-role ARN>"
    ]
```

```
  },
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:TagResource",
    "kms:UntagResource",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

> **Note**
>
> En una política AWS KMS clave, el valor del `Resource` elemento debe ser`*`, lo que significa «esta AWS KMS clave». El asterisco (`*`) identifica la AWS KMS clave a la que se adjunta la política clave.

Concesión de acceso de solo lectura a los metadatos de clave

Para conceder a otros roles de IAM acceso de solo lectura a los metadatos de clave, agregue la instrucción `AllowReadAccessToKeyMetaData` a su política de claves. Por ejemplo, la siguiente declaración le permite enumerar todas las AWS KMS claves de su cuenta con fines de auditoría. Esta declaración concede al usuario AWS raíz acceso de solo lectura a los metadatos clave. Por lo tanto, cualquier entidad principal de IAM en la cuenta puede tener acceso a los metadatos de clave cuando sus políticas basadas en identidad tengan los permisos enumerados en la siguiente instrucción: `kms:Describe*`, `kms:Get*` y `kms:List*`. Asegúrese de reemplazar *`<ID de cuenta>`* por su propia información.

```
{
  "Sid": "AllowReadAcesssToKeyMetaData",
  "Effect": "Allow",
```

```
  "Principal": {
    "AWS": [
      "arn:aws:iam::<accountID>:root"
    ]
  },
  "Action": [
    "kms:Describe*",
    "kms:Get*",
    "kms:List*"
  ],
  "Resource": "*"
}
```

Concesión de permisos de KMS de Amazon SNS a Amazon SNS para publicar mensajes en la cola

Para permitir que su tema de Amazon SNS publique mensajes en su cola cifrada de Amazon SQS, agregue la instrucción de política `AllowSNSToSendToSQS` a su política de claves. Esta declaración otorga a Amazon SNS permisos para usar la AWS KMS clave para publicar en su cola de Amazon SQS. Asegúrese de reemplazar *<ID de cuenta>* por su propia información.

> **Note**
>
> Lo `Condition` indicado en la declaración limita el acceso únicamente al servicio Amazon SNS en la misma AWS cuenta.

```
{
  "Sid": "AllowSNSToSendToSQS",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "sns.amazonaws.com"
    ]
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "<account-id>"
```

```
    }
  }
}
```

Permiso a los consumidores para descifrar mensajes de la cola

La siguiente instrucción `AllowConsumersToReceiveFromTheQueue` concede al consumidor de mensajes de Amazon SQS los permisos necesarios para descifrar los mensajes recibidos de la cola de Amazon SQS cifrada. Cuando adjunte la instrucción de política, reemplace *<ARN del rol de tiempo de ejecución del consumidor>* por el ARN del rol tiempo de ejecución de IAM del consumidor de mensajes.

```
{
  "Sid": "AllowConsumersToReceiveFromTheQueue",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "<consumer's execution role ARN>"
    ]
  },
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Política de Amazon SQS de privilegio mínimo

Esta sección le guía a través de las políticas de colas de Amazon SQS de privilegio mínimo para el caso de uso que se trata en esta guía (por ejemplo, de Amazon SNS a Amazon SQS). La política definida está diseñada para evitar el acceso no intencionado mediante la utilización de una combinación de las instrucciones `Deny` y `Allow`. Las instrucciones `Allow` conceden acceso a la entidad o entidades previstas. Las instrucciones `Deny` impiden que otras entidades no previstas accedan a la cola de Amazon SQS, al mismo tiempo que excluyen a la entidad prevista en la condición de política.

La política de Amazon SQS incluye las siguientes instrucciones, que describimos detalladamente a continuación:

- Restricción de los permisos de administración de Amazon SQS

- Restricción de las acciones de la cola de Amazon SQS de la organización especificada
- Concesión de permisos de Amazon SQS a los consumidores
- Aplicación del cifrado de los datos en tránsito
- Restricción de la transmisión de mensajes a un tema específico de Amazon SNS
- (Opcional) Restricción de la recepción de mensajes a un punto de conexión de VPC específico

Restricción de los permisos de administración de Amazon SQS

La siguiente instrucción de política `RestrictAdminQueueActions` restringe los permisos de administración de Amazon SQS solo al rol o a los roles de IAM que utilice para implementar la cola, administrarla o ambas cosas. Asegúrese de reemplazar los *<valores de marcador de posición>* por su propia información. Especifique el ARN del rol de IAM utilizado para implementar la cola de Amazon SQS, así como los ARN de cualquier rol de administrador que deba tener permisos de administración de Amazon SQS.

```
{
  "Sid": "RestrictAdminQueueActions",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": [
    "sqs:AddPermission",
    "sqs:DeleteQueue",
    "sqs:RemovePermission",
    "sqs:SetQueueAttributes"
  ],
  "Resource": "<SQS Queue ARN>",
  "Condition": {
    "StringNotLike": {
      "aws:PrincipalARN": [
        "arn:aws:iam::<account-id>:role/<deployment-role-name>",
        "<admin-role ARN>"
      ]
    }
  }
}
```

Restricción de las acciones de la cola de Amazon SQS de la organización especificada

Como ayuda para proteger sus recursos de Amazon SQS del acceso externo (acceso por parte de una entidad ajena a su organización de AWS), utilice la siguiente instrucción. Esta instrucción limita el acceso a la cola de Amazon SQS a la organización que especifique en la `Condition`. Asegúrese de reemplazar *<ARN de cola de SQS>* por el ARN del rol de IAM utilizado para implementar la cola de Amazon SQS e *<ID de organización>* por el ID de su organización.

```
{
  "Sid": "DenyQueueActionsOutsideOrg",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": [
    "sqs:AddPermission",
    "sqs:ChangeMessageVisibility",
    "sqs:DeleteQueue",
    "sqs:RemovePermission",
    "sqs:SetQueueAttributes",
    "sqs:ReceiveMessage"
  ],
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "StringNotEquals": {
      "aws:PrincipalOrgID": [
        "<org-id>"
      ]
    }
  }
}
```

Concesión de permisos de Amazon SQS a los consumidores

Para recibir mensajes de la cola de Amazon SQS, debe proporcionar al consumidor de mensajes los permisos necesarios. La siguiente instrucción de política concede al consumidor que especifique los permisos necesarios para consumir mensajes de la cola de Amazon SQS. Al agregar la instrucción a su política de Amazon SQS, asegúrese de reemplazar *<ARN del rol de IAM en tiempo de ejecución del consumidor>* por el ARN del rol de IAM en tiempo de ejecución utilizado por el consumidor y *<ARN de cola de SQS>* por el ARN del rol de IAM utilizado para implementar la cola de Amazon SQS.

```
{
  "Sid": "AllowConsumersToReceiveFromTheQueue",
  "Effect": "Allow",
  "Principal": {
    "AWS": "<consumer's IAM execution role ARN>"
  },
  "Action": [
    "sqs:ChangeMessageVisibility",
    "sqs:DeleteMessage",
    "sqs:GetQueueAttributes",
    "sqs:ReceiveMessage"
  ],
  "Resource": "<SQS queue ARN>"
}
```

Para impedir que otras entidades reciban mensajes de la cola de Amazon SQS, agregue la instrucción `DenyOtherConsumersFromReceiving` a la política de la cola de Amazon SQS. Esta instrucción restringe el consumo de mensajes al consumidor que usted especifique y no permite que otros consumidores tengan acceso, aunque sus permisos de identidad se lo concedan. Asegúrese de reemplazar *<ARN de cola de SQS>* y *<ARN de rol de tiempo de ejecución del consumidor>* por su propia información.

```
{
  "Sid": "DenyOtherConsumersFromReceiving",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": [
    "sqs:ChangeMessageVisibility",
    "sqs:DeleteMessage",
    "sqs:ReceiveMessage"
  ],
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "StringNotLike": {
      "aws:PrincipalARN": "<consumer's execution role ARN>"
    }
  }
}
```

Aplicación del cifrado de los datos en tránsito

La siguiente instrucción de política `DenyUnsecureTransport` obliga a los consumidores y productores a utilizar canales seguros (conexiones TLS) para enviar y recibir mensajes de la cola de Amazon SQS. Asegúrese de reemplazar *<ARN de cola de SQS>* por el ARN del rol de IAM utilizado para implementar la cola de Amazon SQS.

```
{
  "Sid": "DenyUnsecureTransport",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": [
    "sqs:ReceiveMessage",
    "sqs:SendMessage"
  ],
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "Bool": {
      "aws:SecureTransport": "false"
    }
  }
}
```

Restricción de la transmisión de mensajes a un tema específico de Amazon SNS

La siguiente instrucción de política `AllowSNSToSendToTheQueue` permite que el tema de Amazon SNS especificado envíe mensajes a la cola de Amazon SQS. Asegúrese de reemplazar *<ARN de cola de SQS>* por el ARN del rol de IAM utilizado para implementar la cola de Amazon SQS y *<ARN de tema de SNS>* por el ARN de tema de Amazon SNS.

```
{
  "Sid": "AllowSNSToSendToTheQueue",
  "Effect": "Allow",
  "Principal": {
    "Service": "sns.amazonaws.com"
  },
```

```
  "Action": "sqs:SendMessage",
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "<SNS topic ARN>"
    }
  }
}
```

La siguiente instrucción de política `DenyAllProducersExceptSNSFromSending` impide que otros productores envíen mensajes a la cola. Reemplace *<ARN de cola de SQS>* y *<ARN de tema de SNS>* por su propia información.

```
{
  "Sid": "DenyAllProducersExceptSNSFromSending",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": "sqs:SendMessage",
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "ArnNotLike": {
      "aws:SourceArn": "<SNS topic ARN>"
    }
  }
}
```

(Opcional) Restricción de la recepción de mensajes a un punto de conexión de VPC específico

Para restringir la recepción de mensajes a solo un punto de conexión de VPC específico, agregue la siguiente instrucción de política a su política de colas de Amazon SQS. Esta instrucción impide que un consumidor de mensajes reciba mensajes de la cola a menos que los mensajes procedan del punto de conexión de VPC deseado. Reemplace *<ARN de cola de SQS>* por el ARN del rol de IAM utilizado para implementar la cola de Amazon SQS y *<vpce_id>* por el ID del punto de conexión de VPC.

```
{
  "Sid": "DenyReceivingIfNotThroughVPCE",
```

```
  "Effect": "Deny",
  "Principal": "*",
  "Action": [
    "sqs:ReceiveMessage"
  ],
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "StringNotEquals": {
      "aws:sourceVpce": "<vpce id>"
    }
  }
}
```

Instrucciones de política de Amazon SQS para la cola de mensajes fallidos

Agregue las siguientes instrucciones de política, identificadas por su ID de instrucción, a su política de acceso de DLQ:

- `RestrictAdminQueueActions`
- `DenyQueueActionsOutsideOrg`
- `AllowConsumersToReceiveFromTheQueue`
- `DenyOtherConsumersFromReceiving`
- `DenyUnsecureTransport`

Además de agregar las instrucciones de política anteriores a su política de acceso de DLQ, también debe agregar una instrucción para restringir la transmisión de mensajes a las colas de Amazon SQS, como se describe en la sección siguiente.

Restricción de la transmisión de mensajes a las colas de Amazon SQS

Para restringir el acceso solo a las colas de Amazon SQS de la misma cuenta, agregue la siguiente instrucción de política `DenyAnyProducersExceptSQS` a la política de colas de DLQ. Esta instrucción no limita la transmisión de mensajes a una cola específica porque necesita implementar la DLQ antes de crear la cola principal, por lo que no conocerá el ARN de Amazon SQS cuando cree la DLQ. Si necesita limitar el acceso solo a una cola de Amazon SQS, modifique `aws:SourceArn` en la `Condition` con el ARN de su cola de origen de Amazon SQS cuando lo conozca.

```
{
  "Sid": "DenyAnyProducersExceptSQS",
  "Effect": "Deny",
```

```
  "Principal": {
    "AWS": "*"
  },
  "Action": "sqs:SendMessage",
  "Resource": "<SQS DLQ ARN>",
  "Condition": {
    "ArnNotLike": {
      "aws:SourceArn": "arn:aws:sqs:<region>:<account-id>:*"
    }
  }
}
```

> **Important**
>
> Las políticas de colas de Amazon SQS definidas en esta guía no restringen la acción `sqs:PurgeQueue` a un rol o roles de IAM determinados. La acción `sqs:PurgeQueue` le permite eliminar todos los mensajes de la cola de Amazon SQS. También puede utilizar esta acción para realizar cambios en el formato de los mensajes sin reemplazar la cola de Amazon SQS. Al depurar una aplicación, puede borrar la cola de Amazon SQS para eliminar mensajes potencialmente erróneos. Cuando pruebe la aplicación, puede dirigir un elevado volumen de mensajes a través de la cola de Amazon SQS y, a continuación, purgarla para empezar de cero antes de pasar a producción. La razón de no restringir esta acción a un rol determinado se debe a que es posible que este rol no se conozca al implementar la cola de Amazon SQS. Deberá agregar este permiso a la política basada en identidades del rol para poder purgar la cola.

Prevención del problema del suplente confuso entre servicios

El problema del suplente confuso es una cuestión de seguridad en la que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. Para evitarlo, AWS proporciona herramientas que te ayudan a proteger tu cuenta si permites que terceros (lo que se conoce como cuentas múltiples) u otros AWS servicios (lo que se conoce como servicio cruzado) accedan a los recursos de tu cuenta. Las instrucciones de política de esta sección pueden ayudarlo a prevenir el problema del suplente confuso entre servicios.

La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos

de otro cliente de una manera en la que no debe tener permiso para acceder. Como ayuda para protegerse contra este problema, las políticas basadas en recursos definidas en esta publicación utilizan las claves de contexto de condición IAM globales `aws:SourceArn`, `aws:SourceAccount` y `aws:PrincipalOrgID`. Esto limita los permisos que tiene un servicio a un recurso específico, una cuenta específica o una organización específica en AWS Organizations.

Uso de Analizador de acceso de IAM para revisar el acceso entre cuentas

Puede utilizar AWS IAM Access Analyzer para revisar sus políticas de colas AWS KMS y políticas clave de Amazon SQS y avisarle cuando una cola o clave de Amazon SQS conceda acceso a AWS KMS una entidad externa. Analizador de acceso de IAM le ayuda a identificar los recursos de su organización y sus cuentas que se comparten con una entidad externa. Esta zona de confianza puede ser una AWS cuenta o la organización dentro de AWS Organizations que especifique al activar IAM Access Analyzer.

IAM Access Analyzer identifica los recursos compartidos con entidades externas mediante el uso de un razonamiento basado en la lógica para analizar las políticas basadas en los recursos de su entorno. AWS Para cada instancia de un recurso compartido fuera de su zona de confianza, Analizador de acceso genera un resultado. Los resultados incluyen información sobre el acceso y la entidad principal externa a la que se le concede. Revise los resultados para determinar si el acceso es intencionado y seguro, o si el acceso es no intencionado y supone un riesgo para la seguridad. En caso de accesos no intencionados, revise la política afectada y corríjala. Consulte esta entrada del blog para obtener más información sobre cómo AWS IAM Access Analyzer identifica el acceso no deseado a sus recursos. AWS

Para obtener más información sobre AWS IAM Access Analyzer, consulte la documentación de IAM Access Analyzer.AWS

Permisos de la API de Amazon SQS: referencia de acciones y recursos

A la hora de configurar Control de acceso y escribir políticas de permisos que puede asociar a una identidad de IAM, puede utilizar la siguiente tabla como referencia. La incluye cada acción de Amazon Simple Queue Service, las acciones correspondientes para las que puede conceder permisos para realizar la acción y el AWS recurso para el que puede conceder los permisos.

Especifique las acciones en el campo `Action` de la política y el valor del recurso en el campo `Resource` de la política. Para especificar una acción, use el prefijo `sqs:` seguido del nombre de acción (por ejemplo, `sqs:CreateQueue`).

En la actualidad, Amazon SQS admite las claves de contexto de condición global disponibles en IAM.

API de Amazon Simple Queue Service y permisos requeridos para las acciones

AddPermission

Acciones: `sqs:AddPermission`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

ChangeMessageVisibilidad

Acciones: `sqs:ChangeMessageVisibility`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

ChangeMessageVisibilityBatch

Acciones: `sqs:ChangeMessageVisibilityBatch`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

CreateQueue

Acciones: `sqs:CreateQueue`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

DeleteMessage

Acciones: `sqs:DeleteMessage`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

DeleteMessageBatch

Acciones: `sqs:DeleteMessageBatch`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

DeleteQueue

Acciones: `sqs:DeleteQueue`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

GetQueueAtributos

Acciones: `sqs:GetQueueAttributes`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

GetQueueUrl

Acciones: `sqs:GetQueueUrl`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

ListDeadLetterSourceColas

Acciones: `sqs:ListDeadLetterSourceQueues`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

ListQueues

Acciones: `sqs:ListQueues`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

ListQueueEtiquetas

Acciones: `sqs:ListQueueTags`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

PurgeQueue

Acciones: `sqs:PurgeQueue`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

ReceiveMessage

Acciones: `sqs:ReceiveMessage`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

RemovePermission

Acciones: `sqs:RemovePermission`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

SendMessagey SendMessageBatch

Acciones: `sqs:SendMessage`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

SetQueueAtributos

Acciones: `sqs:SetQueueAttributes`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

TagQueue

Acciones: `sqs:TagQueue`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

UntagQueue

Acciones: `sqs:UntagQueue`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

Registro y monitoreo en Amazon SQS

En esta sección se proporciona información sobre las opciones de registro y supervisión de Amazon SQS, incluida la forma de capturar las llamadas CloudTrail a la API y CloudWatch las métricas para obtener información sobre la actividad y el rendimiento de las colas.

Temas

- Registro de llamadas a la API de Amazon SQS mediante AWS CloudTrail
- Supervisión de las colas de Amazon SQS mediante CloudWatch

Registro de llamadas a la API de Amazon SQS mediante AWS CloudTrail

Amazon SQS está integrado AWS CloudTrail para grabar las llamadas de Amazon SQS de un usuario, función o servicio. AWS CloudTrail captura las llamadas a la API relacionadas con el estándar de Amazon SQS y las colas FIFO como eventos, incluidas las interacciones iniciadas a través de la consola de Amazon SQS, así como mediante programación mediante llamadas a las API de Amazon SQS.

Temas

- Información de Amazon SQS en CloudTrail
- Gestión de eventos en CloudTrail
- Eventos de datos en CloudTrail
- Ejemplos: eventos CloudTrail de administración para Amazon SQS
- Ejemplos: eventos CloudTrail de datos para Amazon SQS

Información de Amazon SQS en CloudTrail

CloudTrail está activado de forma predeterminada al crear la cuenta. AWS Cuando se produce una actividad de eventos de Amazon SQS compatible, se registra en un CloudTrail evento, junto con otros eventos de AWS servicio, en el historial de eventos. Puede ver, buscar y descargar los eventos recientes de su AWS cuenta. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos en la Guía del AWS CloudTrail usuario.

Las API de Amazon SQS que llaman a las operaciones de administración de colas, por ejemplo, `AddPermission` se clasifican como eventos de administración y se registran de forma predeterminada. CloudTrail Las API de Amazon SQS que son operaciones de gran volumen que se realizan en una cola de Amazon SQS, por ejemplo, se clasifican `SendMessage` como eventos de datos y se registran una vez que usted se suscribe a ellas. CloudTrail

Con la información CloudTrail recopilada, puede identificar una solicitud específica a una API de Amazon SQS, la dirección IP o identidad del solicitante y la fecha y hora de la solicitud. Si configura una CloudTrail ruta, puede entregar CloudTrail eventos de forma continua a un bucket de Amazon S3 con una entrega opcional a Amazon CloudWatch Logs y AWS EventBridge. Si no configura un registro, solo podrá ver el historial de eventos de los eventos de administración incluidos en los eventos de la CloudTrail consola. Para obtener más información, consulte Información general acerca de la creación de registros de seguimiento en la Guía del usuario de AWS CloudTrail.

Gestión de eventos en CloudTrail

Amazon SQS registra las siguientes acciones de la API como eventos de administración:

- `AddPermission`
- `CreateQueue`
- `CancelMessageMoveTask`
- `DeleteQueue`
- `ListMessageMoveTasks`
- `PurgeQueue`
- `RemovePermission`
- `SetQueueAttributes`
- `StartMessageMoveTask`
- `TagQueue`

- UntagQueue

Las siguientes API de Amazon SQS no se admiten para CloudTrail el registro:

- GetQueueAttributes
- GetQueueUrl
- ListDeadLetterSourceQueues
- ListQueueTags
- ListQueues

Eventos de datos en CloudTrail

Los eventos de datos proporcionan información sobre las operaciones de recursos realizadas en un recurso, como enviar o recibir un mensaje de Amazon SQS hacia y desde una cola de Amazon SQS. Los eventos de datos son actividades de gran volumen que CloudTrail no se registran de forma predeterminada. Puede habilitar el registro de acciones de la API de eventos de datos para su cola de SQS mediante las API. CloudTrail Para obtener más información, consulte Registro de eventos de datos en la Guía del usuario de AWS CloudTrail .

Con CloudTrail, puede utilizar selectores de eventos avanzados para decidir qué actividades de la API de Amazon SQS se registran y registran. Para registrar los eventos de datos de Amazon SQS, debe incluir el tipo de recurso `AWS::SQS::Queue`. Una vez establecido esto, puede ajustar aún más las preferencias de registro seleccionando eventos de datos específicos para registrarlos, por ejemplo, mediante el uso del filtro `eventName` para realizar un seguimiento de los eventos de `SendMessage`. Para obtener más información, consulte `AdvancedEventSelector` en la Referencia de la API de AWS CloudTrail .

Eventos de datos de Amazon SQS:

- SendMessage
- SendMessageBatch
- ReceiveMessage
- DeleteMessage
- DeleteMessageBatch
- ChangeMessageVisibility

- `ChangeMessageVisibilityBatch`

Se aplican cargos adicionales a los eventos de datos. Para más información, consulte Precios de AWS CloudTrail.

Ejemplos: eventos CloudTrail de administración para Amazon SQS

Los siguientes ejemplos muestran las entradas de CloudTrail registro de las API compatibles:

AddPermission

El siguiente ejemplo muestra una entrada de CloudTrail registro para una llamada a la `AddPermission` API.


```
{
   "Records": [
     {
       "eventVersion": "1.06",
       "userIdentity": {
         "type": "IAMUser",
         "principalId": "AKIAI44QH8DHBEXAMPLE",
         "arn": "arn:aws:iam::123456789012:user/Alice",
         "accountId": "123456789012",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "Alice"
       },
       "eventTime": "2018-06-28T22:23:46Z",
       "eventSource": "sqs.amazonaws.com",
       "eventName": "AddPermission",
       "awsRegion": "us-east-2",
       "sourceIPAddress": "203.0.113.0",
       "userAgent": "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101
 Firefox/24.0",
       "requestParameters": {
         "actions": [
           "SendMessage"
         ],
         "AWSAccountIds": [
           "123456789012"
         ],
         "label": "MyLabel",
         "queueUrl": "https://sqs.us-east-2.amazon.com/123456789012/MyQueue"
```

```
      },
      "responseElements": null,
      "requestID": "123abcde-f4gh-50ij-klmn-60o789012p30",
      "eventID": "0987g654-32f1-09e8-d765-c4f3fb2109fa"
    }
  ]
}
```

CreateQueue

El siguiente ejemplo muestra una entrada de CloudTrail registro para una llamada a la `CreateQueue` API.

```
{
   "Records": [
     {
       "eventVersion": "1.06",
       "userIdentity": {
         "type": "IAMUser",
         "principalId": "AKIAI44QH8DHBEXAMPLE",
         "arn": "arn:aws:iam::123456789012:user/Alejandro",
         "accountId": "123456789012",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "Alejandro"
       },
       "eventTime": "2018-06-28T22:23:46Z",
       "eventSource": "sqs.amazonaws.com",
       "eventName": "CreateQueue",
       "awsRegion": "us-east-2",
       "sourceIPAddress": "203.0.113.1",
       "userAgent": "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101
 Firefox/24.0",
       "requestParameters": {
         "queueName": "MyQueue"
       },
       "responseElements": {
         "queueUrl": "https://sqs.us-east-2.amazon.com/123456789012/MyQueue"
       },
       "requestID": "123abcde-f4gh-50ij-klmn-60o789012p30",
       "eventID": "0987g654-32f1-09e8-d765-c4f3fb2109fa"
     }
   ]
}
```

DeleteQueue

El siguiente ejemplo muestra una entrada de CloudTrail registro para una llamada a la `DeleteQueue` API.


```
{
   "Records": [
     {
       "eventVersion": "1.06",
       "userIdentity": {
         "type": "IAMUser",
         "principalId": "AKIAI44QH8DHBEXAMPLE",
         "arn": "arn:aws:iam::123456789012:user/Carlos",
         "accountId": "123456789012",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "Carlos"
       },
       "eventTime": "2018-06-28T22:23:46Z",
       "eventSource": "sqs.amazonaws.com",
       "eventName": "DeleteQueue",
       "awsRegion": "us-east-2",
       "sourceIPAddress": "203.0.113.2",
       "userAgent": "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101
 Firefox/24.0",
       "requestParameters": {
         "queueUrl": "https://sqs.us-east-2.amazon.com/123456789012/MyQueue"
       },
       "responseElements": null,
       "requestID": "123abcde-f4gh-50ij-klmn-60o789012p30",
       "eventID": "0987g654-32f1-09e8-d765-c4f3fb2109fa"
     }
   ]
 }
```


RemovePermission

El siguiente ejemplo muestra una entrada de CloudTrail registro para una llamada a la `RemovePermission` API.


```
{
   "Records": [
     {
       "eventVersion": "1.06",
```

```
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Jane",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Jane"
      },
      "eventTime": "2018-06-28T22:23:46Z",
      "eventSource": "sqs.amazonaws.com",
      "eventName": "RemovePermission",
      "awsRegion": "us-east-2",
      "sourceIPAddress": "203.0.113.3",
      "userAgent": "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101
 Firefox/24.0",
      "requestParameters": {
        "label": "label",
        "queueUrl": "https://sqs.us-east-2.amazon.com/123456789012/MyQueue"
      },
      "responseElements": null,
      "requestID": "123abcde-f4gh-50ij-klmn-60o789012p30",
      "eventID": "0987g654-32f1-09e8-d765-c4f3fb2109fa"
    }
  ]
 }
```

SetQueueAttributes

El siguiente ejemplo muestra una entrada de CloudTrail registro para`SetQueueAttributes`:

```
{
   "Records": [
     {
       "eventVersion": "1.06",
       "userIdentity": {
         "type": "IAMUser",
         "principalId": "AKIAI44QH8DHBEXAMPLE",
         "arn": "arn:aws:iam::123456789012:user/Maria",
         "accountId": "123456789012",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "Maria"
       },
       "eventTime": "2018-06-28T22:23:46Z",
       "eventSource": "sqs.amazonaws.com",
```

```
      "eventName": "SetQueueAttributes",
      "awsRegion": "us-east-2",
      "sourceIPAddress": "203.0.113.4",
      "userAgent": "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101
Firefox/24.0",
      "requestParameters": {
        "attributes": {
          "VisibilityTimeout": "100"
        },
        "queueUrl": "https://sqs.us-east-2.amazon.com/123456789012/MyQueue"
      },
      "responseElements": null,
      "requestID": "123abcde-f4gh-50ij-klmn-60o789012p30",
      "eventID": "0987g654-32f1-09e8-d765-c4f3fb2109fa"
    }
  ]
}
```

Ejemplos: eventos CloudTrail de datos para Amazon SQS

Los siguientes son ejemplos de CloudTrail eventos específicos de las API de eventos de datos de Amazon SQS:

SendMessage

El siguiente ejemplo muestra un evento CloudTrail de datos para`SendMessage`.

```
  {
 "eventVersion": "1.09",
 "userIdentity": {
   "type": "AssumedRole",
   "principalId": "EXAMPLE_PRINCIPAL_ID",
   "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/SessionName",
   "accountId": "123456789012",
   "accessKeyId": "ACCESS_KEY_ID",
   "sessionContext": {
     "sessionIssuer": {
       "type": "Role",
       "principalId": "AKIAI44QH8DHBEXAMPLE",
       "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed",
       "accountId": "123456789012",
```

```
        "userName": "RoleToBeAssumed"
      },
      "attributes": {
        "creationDate": "2023-11-07T22:13:06Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-11-07T23:59:11Z",
  "eventSource": "sqs.amazonaws.com",
  "eventName": "SendMessage",
  "awsRegion": "ap-southeast-4",
  "sourceIPAddress": "10.0.118.80",
  "userAgent": "aws-cli/1.29.16 md/Botocore#1.31.16 ua/2.0 os/
linux#5.4.250-173.369.amzn2int.x86_64 md/arch#x86_64 lang/python#3.8.17 md/
pyimpl#CPython cfg/retry-mode#legacy botocore/1.31.16",
  "requestParameters": {
    "queueUrl": "https://sqs.ap-southeast-4.amazonaws.com/123456789012/MyQueue",
    "messageBody": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "messageDeduplicationId": "MsgDedupIdSdk1ae1958f2-bbe8-4442-83e7-4916e3b035aa",
    "messageGroupId": "MsgGroupIdSdk16"
  },
  "responseElements": {
    "mD5OfMessageBody": "9a4e3f7a614d9dd9f8722092dbda17a2",
    "mD5OfMessageSystemAttributes": "f88f0587f951b7f5551f18ae699c3a9d",
    "messageId": "93bb6e2d-1090-416c-81b0-31eb1faa8cd8",
    "sequenceNumber": "18881790870905840128"
  },
  "requestID": "c4584600-fe8a-5aa3-a5ba-1bc42f055fae",
  "eventID": "98c735d8-70e0-4644-9432-b6ced4d791b1",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::SQS::Queue",
      "ARN": "arn:aws:sqs:ap-southeast-4:123456789012:MyQueue"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "123456789012",
  "eventCategory": "Data",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
```

```
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "sqs.ap-southeast-4.amazonaws.com"
  }
```

ReceiveMessage

En el siguiente ejemplo se muestra un evento de CloudTrail datos para`ReceiveMessage`.

```
    {
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/SessionName",
    "accountId": "123456789012",
    "accessKeyId": "ACCESS_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed",
        "accountId": "123456789012",
        "userName": "RoleToBeAssumed"
      },
      "attributes": {
        "creationDate": "2023-11-07T22:13:06Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-11-07T23:59:24Z",
  "eventSource": "sqs.amazonaws.com",
  "eventName": "ReceiveMessage",
  "awsRegion": "ap-southeast-4",
  "sourceIPAddress": "10.0.118.80",
  "userAgent": "aws-cli/1.29.16 md/Botocore#1.31.16 ua/2.0 os/
linux#5.4.250-173.369.amzn2int.x86_64 md/arch#x86_64 lang/python#3.8.17 md/
pyimpl#CPython cfg/retry-mode#legacy botocore/1.31.16",
  "requestParameters": {
    "queueUrl": "https://sqs.ap-southeast-4.amazonaws.com/123456789012/MyQueue",
    "maxNumberOfMessages": 10
  },
```

```
  "responseElements": null,
  "requestID": "8b4d4643-8f49-52cd-a6e8-1b875ed54b99",
  "eventID": "f3f23ab7-b0a4-4b71-afc0-141209c49206",
  "readOnly": true,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::SQS::Queue",
      "ARN": "arn:aws:sqs:ap-southeast-4:123456789012:MyQueue"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "123456789012",
  "eventCategory": "Data",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "sqs.ap-southeast-4.amazonaws.com"
  }
}
```

DeleteMessageBatch

En el siguiente ejemplo se muestra un evento de CloudTrail datos para`DeleteMessageBatch`.

```
    {
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/SessionName",
    "accountId": "123456789012",
    "accessKeyId": "ACCESS_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed",
        "accountId": "123456789012",
        "userName": "RoleToBeAssumed"
      },
```

```
      "attributes": {
        "creationDate": "2023-11-07T22:13:06Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-11-07T23:59:24Z",
  "eventSource": "sqs.amazonaws.com",
  "eventName": "DeleteMessageBatch",
  "awsRegion": "ap-southeast-4",
  "sourceIPAddress": "10.0.118.80",
  "userAgent": "aws-cli/1.29.16 md/Botocore#1.31.16 ua/2.0 os/
linux#5.4.250-173.369.amzn2int.x86_64 md/arch#x86_64 lang/python#3.8.17 md/
pyimpl#CPython cfg/retry-mode#legacy botocore/1.31.16",
  "requestParameters": {
    "queueUrl": "https://sqs.ap-southeast-4.amazonaws.com/123456789012/MyQueue",
    "entries": [
      {
        "id": "0",
        "receiptHandle": "AQEBefxM1O4zyZGF87DehbRbmri91w2W7mMdD0GrBjQa8e/
hpb4RbXHPZ9tLBVleECbChQIE5NtaDuoZhZPOkTy0eN46EyRR4jXDzE3AlkbPlX1mA9f2fUuTrXx8aeCoCA3I3woNg3f
hlLS94tjAZqV2krc4BaC2pYgjyHWcW019HwIV8T/bjNMIeZoQwOM5V
+o9vHPfewz5QGr5SKpDo7uE7Umyk5n5CJZvcn1efp/
mrwtaCIb9M7cCQUYcZm2ZmZDnIO9XpGTAi3m2dQ0M83pnNh0nvDfpkHpoa+hX1TrUmxCupCWHJwA8HFJ1O/
CCJsodMNFthLBA9S57dkBZCsw41G8jAmgQ0MkvZ0UL5mg0OFQQd1Yrw0zvthjCgiwdzn0yXoMzxIZMBxkY14E4nVVZ7N
h8oRk2C7gByzg2kYJ0LnUvLJFT8DQE28JZppEC9klvrdR/BWiPT7asc="
      }
    ]
  },
  "responseElements": {
    "successful": [
      {
        "id": "0"
      }
    ],
    "failed": []
  },
  "requestID": "fe423091-5642-5ba5-9256-6d5587de52f1",
  "eventID": "88c8020d-d769-4985-8ecb-ee0b59acc418",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::SQS::Queue",
```

```
      "ARN": "arn:aws:sqs:ap-southeast-4:123456789012:MyQueue"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "123456789012",
  "eventCategory": "Data",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "sqs.ap-southeast-4.amazonaws.com"
  }
}
```

ChangeMessageVisibilityBatch

En el siguiente ejemplo se muestra un evento de CloudTrail datos para`ChangeMessageVisibilityBatch`.

```
  {
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/SessionName",
    "accountId": "123456789012",
    "accessKeyId": "ACCESS_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed",
        "accountId": "123456789012",
        "userName": "RoleToBeAssumed"
      },
      "attributes": {
        "creationDate": "2023-11-07T22:13:06Z",
        "mfaAuthenticated": "false"
      }
    }
  },
```

```
  "eventTime": "2023-11-07T23:59:01Z",
  "eventSource": "sqs.amazonaws.com",
  "eventName": "ChangeMessageVisibilityBatch",
  "awsRegion": "ap-southeast-4",
  "sourceIPAddress": "10.0.118.80",
  "userAgent": "aws-cli/1.29.16 md/Botocore#1.31.16 ua/2.0 os/
linux#5.4.250-173.369.amzn2int.x86_64 md/arch#x86_64 lang/python#3.8.17 md/
pyimpl#CPython cfg/retry-mode#legacy botocore/1.31.16",
  "requestParameters": {
    "visibilityTimeout": 0,
    "entries": [
      {
        "id": "0",
        "receiptHandle":
 "AQEB2M5cVYg5gslhWME6537hdjcaPnOYPA5M0W460TTb0DzPle631yPWm8qxd4O1hDj/
B4ntTMnsgBTa95t14tNx7Vn96jKJ5rIoZ7iI8TRmkT1caKodKIPs8w9yndZq50c2FPQxtyH+2L3UHf/
abV3szqVWXOLZR4PwX8zZkWVQGNCNnY2q2lGCG586F8QwvrOFYoXNwB8ymd1t77e1PDPknq1Io3JFuzkEsndkkETy4fV
l5PHX17nXxaC+DURVlMPXOuSFACGmWqAoyk50HKwGOjLQgpySL/
TcnQXClvFq8kNXGwyVzJsbwHpOHxI7oce69vaD6DaWFP75d3hx+PJeG9pauQCKzVP3skt3Hw/
zDC7YfKcALD3aCwMmeNDwT3w0BUG6XZdG5lYhtFtTQYV7YuS3i/
Jh3HShGbtm07JKOEFiPkxv2+XNaAX3gFEpbng6zamTanfyMXCJIiglAEqiyWHQ=",
        "visibilityTimeout": 2271
      }
    ],
    "queueUrl": "https://sqs.ap-southeast-4.amazonaws.com/123456789012/MyQueue"
  },
  "responseElements": {
    "successful": [
      {
        "id": "0"
      }
    ]
  },
  "requestID": "d49ab65f-9dc7-54b8-875c-eb9b4c42988b",
  "eventID": "ca16c8c2-c4ba-4eb5-a54c-e650a10266d4",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::SQS::Queue",
      "ARN": "arn:aws:sqs:ap-southeast-4:123456789012:MyQueue"
    }
  ],
  "eventType": "AwsApiCall",
```

```
  "managementEvent": false,
  "recipientAccountId": "123456789012",
  "eventCategory": "Data",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "sqs.ap-southeast-4.amazonaws.com"
  }
}
```


Supervisión de las colas de Amazon SQS mediante CloudWatch

Amazon SQS y Amazon CloudWatch están integrados para que pueda utilizarlos CloudWatch para ver y analizar las métricas de sus colas de Amazon SQS. Puede ver y analizar las métricas de sus colas desde la consola de Amazon SQS, CloudWatch la consola o mediante AWS CLIla API. CloudWatch También puede configurar CloudWatch alarmas para las métricas de Amazon SQS.

CloudWatch Las métricas de sus colas de Amazon SQS se recopilan automáticamente y se envían a CloudWatch intervalos de un minuto. Estas métricas se recopilan en todas las colas que cumplen con las CloudWatch pautas de actividad. CloudWatchconsidera que una cola está activa durante un máximo de seis horas si contiene algún mensaje o si alguna acción permite acceder a ella.

Cuando una cola de Amazon SQS permanece inactiva durante más de seis horas, el servicio Amazon SQS se considera inactivo y deja de enviar métricas al servicio. CloudWatch Los datos faltantes o los datos que representan cero no se pueden visualizar en las CloudWatch métricas de Amazon SQS durante el período en que la cola de Amazon SQS estuvo inactiva.

> **Note**
>
> - Se puede activar una cola de Amazon SQS cuando el usuario que llama a una API en contra de la cola no está autorizado y se produce un error en la solicitud.
> - La consola Amazon SQS realiza una llamada a la `GetQueueAttributes` API cuando se abre la página de la cola. La solicitud `GetQueueAttributes` de API activa la cola.
> - Cuando una cola se activa desde un estado inactivo, se produce un retraso de hasta 15 minutos en las CloudWatch métricas.
> - No se cobran cargos por las métricas de Amazon SQS publicadas en. CloudWatch Se ofrecen como parte del servicio de Amazon SQS.

- CloudWatch las métricas se admiten tanto para las colas estándar como para las FIFO.

Temas

Acceso a CloudWatch las métricas de Amazon SQS

Amazon SQS y Amazon CloudWatch están integrados para que pueda utilizarlos CloudWatch para ver y analizar las métricas de sus colas de Amazon SQS. Puede ver y analizar las métricas de sus colas desde la consola de Amazon SQS, CloudWatch la consola o mediante AWS CLIla API. CloudWatch También puede configurar CloudWatch alarmas para las métricas de Amazon SQS.

Consola de Amazon SQS

1. Inicie sesión en la consola de Amazon SQS.
2. En la lista de colas, elija (marque) las casillas correspondientes a las colas a cuyas métricas desea obtener acceso. Puede mostrar métricas para un máximo de 10 colas.
3. Elija la pestaña Monitorización.

   En la sección SQS metrics se muestran varios gráficos.
4. Para comprender lo que un determinado gráfico representa, pase el ratón por encima del icono 

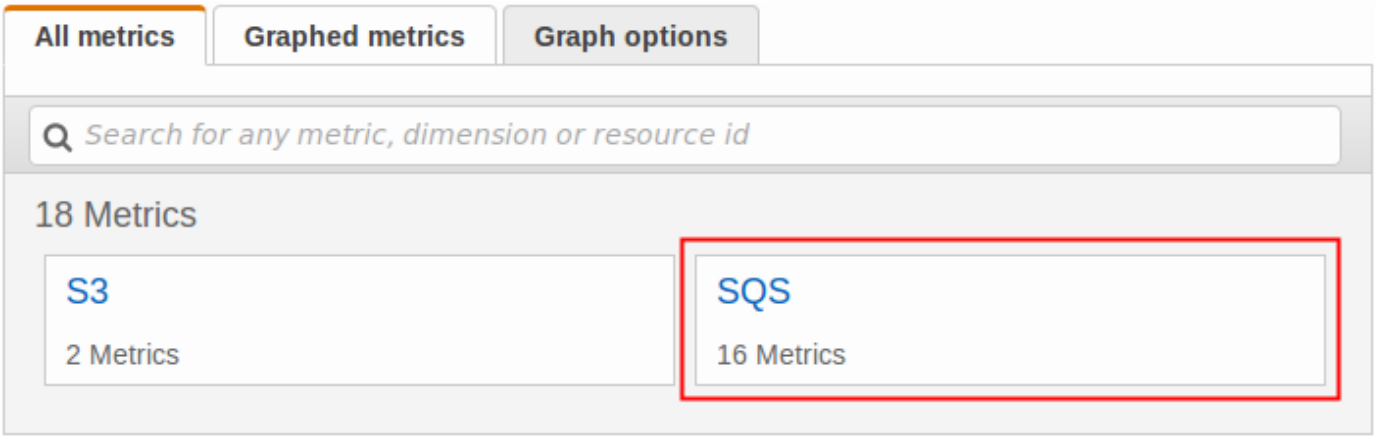
   situado junto al gráfico correspondiente o consulte CloudWatch Métricas disponibles para Amazon SQS.
5. Para cambiar el intervalo de tiempo para todos los gráficos al mismo tiempo, en Time Range, elija el intervalo de tiempo deseado (por ejemplo, Last Hour).
6. Para ver estadísticas adicionales para un gráfico individual, elija el gráfico.
7. En el cuadro de diálogo Detalles de CloudWatch supervisión, seleccione una estadística (por ejemplo, Suma). Para ver una lista de las estadísticas admitidas, consulte CloudWatch Métricas disponibles para Amazon SQS.
8. Para cambiar el intervalo de tiempo durante el cual se muestra un gráfico individual (por ejemplo, para mostrar un intervalo de tiempo de las últimas 24 horas en lugar de los últimos cinco

minutos, o para mostrar un periodo de tiempo de cada hora en lugar de cada 5 minutos), con el cuadro de diálogo del gráfico todavía abierto, en Time Range, elija el intervalo de tiempo deseado (por ejemplo, Last 24 Hours). En Period, elija el periodo de tiempo deseado dentro del intervalo de tiempo especificado (por ejemplo, 1 Hour). Cuando termine de examinar el gráfico, elija Close.

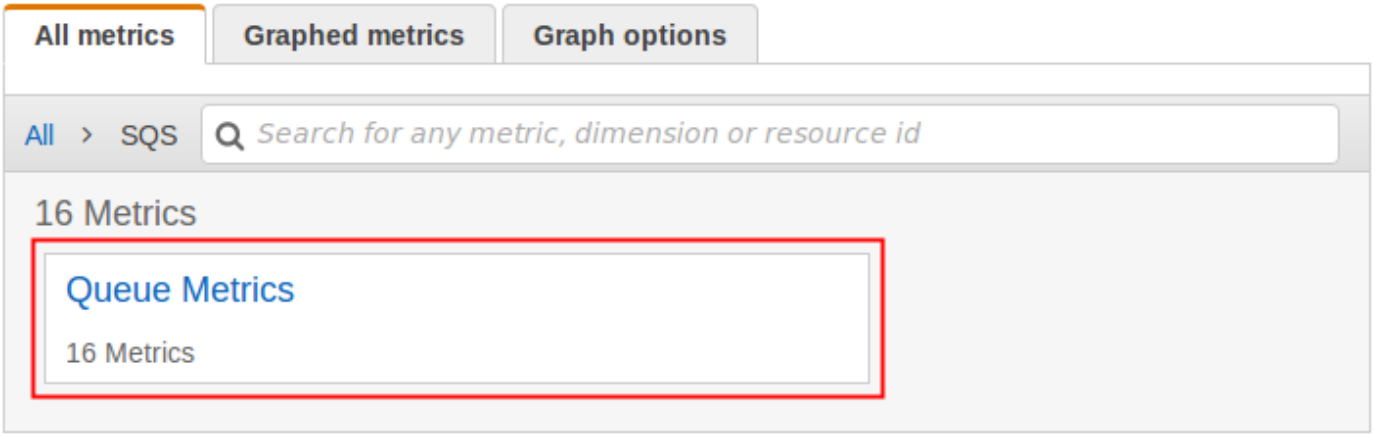
9. (Opcional) Para trabajar con CloudWatch funciones adicionales, en la pestaña Supervisión, elija Ver todas las CloudWatch métricas y, a continuación, siga las instrucciones del CloudWatch Consola Amazon procedimiento.

CloudWatch Consola Amazon

1. Inicie sesión en la consola de CloudWatch.
2. En el panel de navegación, elija Metrics.
3. Seleccione el espacio de nombres de métricas SQS.

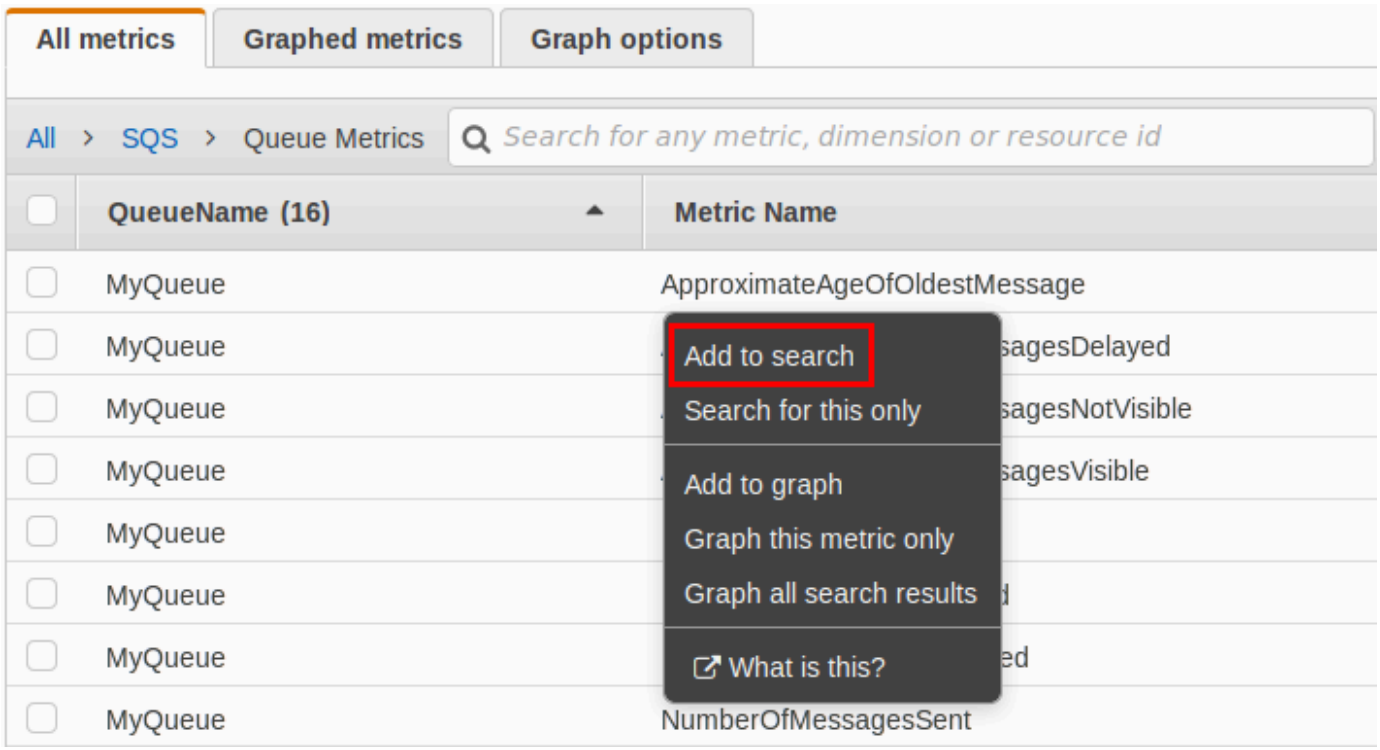


4. Seleccione la dimensión de métricas Queue Metrics.



5. Ahora puede examinar las métricas de Amazon SQS:

- Para ordenar las métricas, utilice el encabezado de columna.
- Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella.
- Para filtrar por métrica, elija el nombre de la métrica y, a continuación, seleccione Add to search (Añadir a búsqueda).



Para obtener más información y opciones adicionales, consulte Graph Metrics and Using Amazon CloudWatch Dashboards en la Guía del CloudWatch usuario de Amazon.

AWS Command Line Interface

Para acceder a las métricas de Amazon SQS mediante AWS CLI, ejecute el `get-metric-statistics` comando.

Para obtener más información, consulta Obtener estadísticas de una métrica en la Guía del CloudWatch usuario de Amazon.

CloudWatch API

Para acceder a las métricas de Amazon SQS mediante la CloudWatch API, utilice la `GetMetricStatistics` acción.

Para obtener más información, consulta Obtener estadísticas de una métrica en la Guía del CloudWatch usuario de Amazon.

Creación de CloudWatch alarmas para las métricas de Amazon SQS

CloudWatch permite activar alarmas en función de un umbral métrico. Por ejemplo, puede crear una alarma para la métrica `NumberOfMessagesSent`. Por ejemplo, si se envían más de 100 mensajes a la cola `MyQueue` en una hora, se envía una notificación por correo electrónico. Para obtener más información, consulte Creación de CloudWatch alarmas de Amazon en la Guía del CloudWatch usuario de Amazon.

1. Inicie sesión en la CloudWatch consola AWS Management Console y ábrala en https://console.aws.amazon.com/cloudwatch/.
2. Elija Alarms y, a continuación, seleccione Create Alarm.
3. En la sección Select Metric (Seleccionar métrica) del cuadro de diálogo Create Alarm (Crear alarma), elija Browse Metrics (Examinar métricas), SQS.
4. En SQS > Queue Metrics, elija el nombre QueueNamey el nombre de la métrica para los que desee configurar una alarma y, a continuación, elija Siguiente. Para ver una lista de las métricas disponibles, consulte CloudWatch Métricas disponibles para Amazon SQS.

   En el siguiente ejemplo, la selección que se realiza corresponde a una alarma de la métrica `NumberOfMessagesSent` para la cola `MyQueue`. La alarma se activa cuando el número de mensajes enviados supera los 100.
5. En la sección Define Alarm (Definir alarma) del cuadro de diálogo Create Alarm (Crear alarma), haga lo siguiente:

   a. En Alarm Threshold (Umbral de alarma), escriba el nombre de la alarma en Name (Nombre) y la descripción en Description (Descripción).

   b. Establezca is en > 100.

   c. Establezca for (para) en 1 out of 1 datapoints (1 de 1 punto de datos).

   d. En Alarm preview (Vista previa de alarma), establezca el valor de Period (Período) en 1 Hour (1 hora).

   e. Establezca Statistic (Estadística) en Standard (Estándar), Sum (Suma).

   f. En Actions (Acciones), establezca Whenever this alarm (Siempre que esta alarma) en State is ALARM (El estado es ALARM).

Si desea CloudWatch enviar una notificación cuando se active la alarma, seleccione un tema existente de Amazon SNS o elija Nueva lista e introduzca las direcciones de correo electrónico separadas por comas.

> **Note**
>
> Si crea un nuevo tema de Amazon SNS, debe verificar las direcciones de correo electrónico antes de que puedan recibir notificaciones. Si el estado de la alarma cambia antes de que se verifiquen las direcciones de correo electrónico, las notificaciones no se envían.

6. Seleccione Crear alarma.

   Se crea la alarma.

CloudWatch Métricas disponibles para Amazon SQS

Amazon SQS envía las siguientes métricas a. CloudWatch

> **Note**
>
> En el caso de las colas estándar, el resultado es aproximado debido a la arquitectura distribuida de Amazon SQS. En la mayoría de los casos, el recuento debe aproximarse al número real de mensajes de la cola.
> Para colas FIFO, el resultado es exacto.

Métricas de Amazon SQS

El espacio de nombres de `AWS/SQS` incluye las siguientes métricas.

Métrica	Descripción
`ApproximateAgeOfOldestMessage`	La antigüedad aproximada del mensaje no eliminado más antiguo de la cola.

Métrica	Descripción
	Note • Si el mensaje se recibe tres veces (o más) y no se procesa, se devuelve a la cola y la métrica `ApproximateAgeOfOldestMessage` apunta al segundo mensaje más antiguo que no se haya recibido más de tres veces. Esta acción se produce aunque la cola tenga una directiva de redireccionamiento. • Dado que un solo mensaje de píldora venenosa (que se ha recibido varias veces, pero no se ha eliminado) puede distorsionar esta métrica, no se incluye la antigüedad de los mensajes de píldora venenosa en la métrica hasta que el mensaje se procesa correctamente. • Cuando la cola tiene una política de redireccionamiento, el mensaje se mueve a una cola de mensajes fallidos después del número máximo de recepciones establecido. Cuando el mensaje se mueve a la cola de mensajes fallidos, la métrica `Approxima`

Métrica	Descripción
	`teAgeOfOldestMessage` de la cola de mensajes fallidos representa la hora a la que el mensaje se trasladó a la cola de mensajes fallidos (no la hora original en la que se envió el mensaje). • Para las colas FIFO, el mensaje no se traslada al final de la cola porque se rompería la garantía de orden FIFO. En su lugar, el mensaje irá a la DLQ si hay una configurada. De lo contrario, bloqueará el grupo de mensajes hasta que se elimine correctamente o hasta que caduque. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: segundos Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS)

Métrica	Descripción
`ApproximateNumberOfMessagesDelayed`	El número de mensajes de la cola que van con retraso y no están disponibles para su lectura inmediata. Esto puede ocurrir cuando la cola está configurada como una cola de retraso o cuando se ha enviado un mensaje con un parámetro de retraso. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: recuento Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS)
`ApproximateNumberOfMessagesNotVisible`	El número de mensajes que se encuentran en tránsito. Se considera que los mensajes están en tránsito si se han enviado a un cliente pero aún no se han eliminado o aún no han llegado al final de su periodo de visibilidad. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: recuento Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS)

Métrica	Descripción
`ApproximateNumberOfMessagesVisible`	El número de mensajes que se procesará n. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: recuento Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS) No hay límite en el número de mensajes para procesar; no obstante, puede someter estas tareas pendientes a un periodo de retención.
`NumberOfEmptyReceives` [1]	El número de llamadas a la API `ReceiveMessage` que no devolvieron un mensaje. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: recuento Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS)

Métrica	Descripción
`NumberOfMessagesDeleted` [1]	El número de mensajes eliminados de la cola. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: recuento Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS) Amazon SQS emite la métrica `NumberOfMessagesDeleted` para cada operación de eliminación realizada correctamente que use un identificador de recepción válido, incluidas las operaciones de eliminación duplicadas. Las siguientes situaciones pueden provocar que el valor de la métrica `NumberOfMessagesDeleted` sea superior al esperado: • Llamar a la acción `DeleteMessage` en identificadores de recepción diferentes que pertenecen al mismo mensaje: si el mensaje no se procesa antes de que se agote el tiempo de espera de visibilidad, queda a disposición de otros consumidores que pueden procesarlo y volver a eliminarlo, lo que incrementa el valor de la métrica `NumberOfMessagesDeleted` . •

Métrica	Descripción
	Llamar a la acción `DeleteMessage` en el mismo identificador de recepción : si el mensaje se procesa y elimina pero se llama de nuevo a la acción `DeleteMessage` usando el mismo indicador de recepción, se devuelve un estado de éxito, lo que increment a el valor de la métrica `NumberOfMessagesDeleted` .
`NumberOfMessagesReceived` [1]	El número de mensajes devueltos por llamadas a la acción de la `ReceiveMessage` . Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: recuento Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS)

Métrica	Descripción
`NumberOfMessagesSent` [1]	Número de mensajes añadidos a una cola. Si envía manualmente un mensaje a una cola de mensajes fallidos, la métrica `NumberOfMessagesSent` lo captura. Sin embargo, si un mensaje se envía a una cola de letra muerta como resultado de un intento de procesamiento fallido, esta métrica no lo captura. Por tanto, es posible que los valores de `NumberOfMessagesSent` y `NumberOfMessagesReceived` sean diferentes. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: recuento Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS)

Métrica	Descripción
`SentMessageSize` [1]	Tamaño de los mensajes añadidos a una cola. Criterios de notificación: se informa de un valor no negativo si la cola está activa. Unidades: bytes Estadísticas válidas: Promedio, Mínimo, Máximo, Suma, Muestras de datos (se muestra como Número de muestras en la consola de Amazon SQS) **Note** `SentMessageSize` no se muestra como métrica disponible en la CloudWatch consola hasta que se envíe al menos un mensaje a la cola correspondiente.

[1] Estas métricas se calculan desde la perspectiva del servicio y pueden incluir reintentos. No confíe en los valores absolutos de estas métricas ni las utilice para realizar una estimación del estado actual de las colas.

Dimensiones para las métricas de Amazon SQS

La única dimensión a la que Amazon SQS envía es. CloudWatch `QueueName` Esto significa que todas las estadísticas disponibles se filtran por `QueueName`.

Validación de la conformidad de Amazon SQS

Para saber si una Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento

Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- Diseño de arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar AWS las empresas para crear aplicaciones aptas para la HIPAA.

  > **Note**
  >
  > No Servicios de AWS todas cumplen con los requisitos de la HIPAA. Para más información, consulte la Referencia de servicios compatibles con HIPAA.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.

- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en Amazon SQS

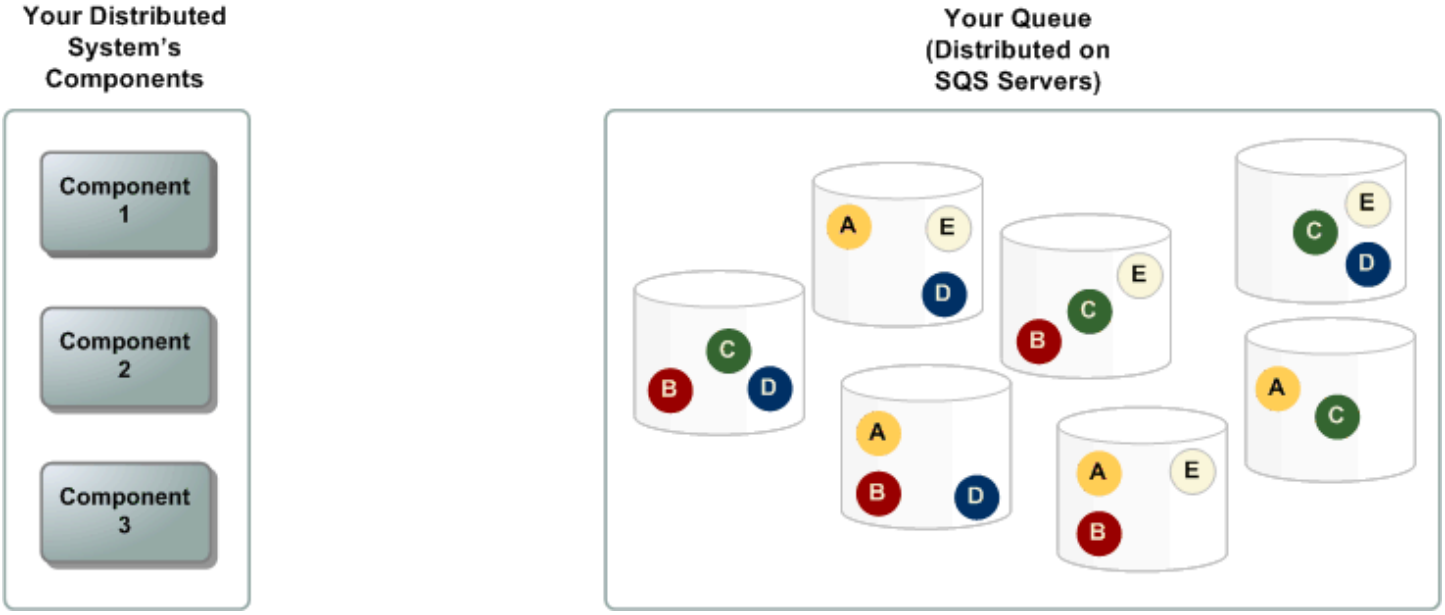
La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos. Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte AWS Infraestructura global.

Además de la infraestructura AWS global, Amazon SQS ofrece colas distribuidas.

Colas distribuidas

Un sistema de mensajería distribuido consta de tres partes principales: los componentes del sistema distribuido, la cola (distribuida en los servidores de Amazon SQS) y los mensajes de la cola.

En el siguiente escenario, el sistema tiene varios productores (componentes que envían mensajes a la cola) y consumidores (componentes que reciben mensajes de la cola). La cola (que contiene los mensajes A a E) almacena de forma redundante los mensajes en varios servidores de Amazon SQS.



Seguridad de la infraestructura en Amazon SQS

Como servicio gestionado, Amazon SQS está protegido por los procedimientos de seguridad de red AWS global descritos en el documento técnico Amazon Web Services: Overview of Security Processes.

Las acciones de API AWS publicadas se utilizan para acceder a Amazon SQS a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE).

Debe firmar las solicitudes mediante un ID de clave de acceso y una clave de acceso secreta asociada a una entidad principal de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puede llamar a estas acciones de la API desde cualquier ubicación de red, pero Amazon SQS admite políticas de acceso basadas en recursos, que pueden incluir restricciones en función de la dirección IP del origen. También puede utilizar políticas de Amazon SQS para controlar el acceso desde puntos de conexión de Amazon VPC específicos o VPC específicas. Esto aísla eficazmente el acceso a la red a una cola de Amazon SQS determinada únicamente de la VPC específica de la red. AWS Para obtener más información, consulte Ejemplo 5: denegar el acceso si no es desde un punto de enlace de la VPC.

Prácticas recomendadas de seguridad para Amazon SQS

AWS proporciona numerosas funciones de seguridad para Amazon SQS, que debe revisar en el contexto de su propia política de seguridad. A continuación, se indican las prácticas recomendadas de seguridad preventiva para Amazon SQS.

> **Note**
>
> La orientación de implementación específica que se proporciona corresponde a casos de uso e implementaciones habituales. Le sugerimos que consulte estas prácticas recomendadas en el contexto de su caso de uso, arquitectura y modelo de amenaza concretos.

Temas

Comprobación de que las colas no sean accesibles de forma pública

A menos que exija explícitamente a cualquier persona de Internet que pueda leer o escribir en su cola de Amazon SQS, debe asegurarse de que su cola no sea de acceso público (a la que puedan acceder todos los habitantes del mundo o cualquier usuario autenticado). AWS

- Evite la creación de políticas con `Principal` establecido en `""`.
- Evite usar un carácter comodín (`*`). En su lugar, designe a un usuario o usuarios específicos.

Implementación del acceso a los privilegios mínimos

Cuando concede permisos, decide quién los recibe, para qué colas son los permisos y las acciones específicas de la API que desea permitir en estas colas. La implementación de los privilegios

mínimos es importante para reducir los riesgos de seguridad y disminuir el efecto de errores o intenciones maliciosas.

Siga el consejo de seguridad estándar de concesión del privilegio mínimo. Es decir, conceda solo los permisos necesarios para realizar una tarea específica. Puede efectuar esta impresora con una combinación de políticas de seguridad.

Amazon SQS utiliza el modelo productor-consumidor, que requiere tres tipos de acceso a la cuenta de usuario:

- Administradores: acceso a la creación, modificación y eliminación de colas. Los administradores también controlan las políticas de cola.
- Productores: acceso al envío de mensajes a las colas.
- Consumidores: acceso a la recepción y eliminación de mensajes de las colas.

Para obtener más información, consulte las siguientes secciones:

- Identity and Access Management en Amazon SQS
- Permisos de la API de Amazon SQS: referencia de acciones y recursos
- Uso de políticas personalizadas con el lenguaje de la política de acceso de Amazon SQS

Utilice funciones de IAM para aplicaciones y AWS servicios que requieren acceso a Amazon SQS

Para que aplicaciones o AWS servicios como Amazon EC2 puedan acceder a las colas de Amazon SQS, deben utilizar credenciales AWS válidas en sus solicitudes de API. AWS Como estas credenciales no se rotan automáticamente, no debe almacenar las AWS credenciales directamente en la aplicación o en la instancia de EC2.

Debe utilizar un rol de IAM para administrar de manera temporal credenciales para las aplicaciones o los servicios que necesiten acceder a Amazon SQS. Cuando utiliza un rol, no tiene que distribuir credenciales de larga duración (como un nombre de usuario, una contraseña y claves de acceso) a una instancia o AWS servicio de EC2 como. AWS Lambda En cambio, el rol proporciona permisos temporales que las aplicaciones pueden usar cuando realizan llamadas a otros AWS recursos.

Para obtener más información, consulte Roles de IAM y Situaciones habituales con los roles: usuarios, aplicaciones y servicios en la Guía del usuario de IAM.

Implementación del cifrado en el servidor

Para mitigar los problemas de fuga de datos, utilice el cifrado en reposo para cifrar sus mensajes mediante una clave almacenada en una ubicación distinta de la ubicación en la que se almacenan los mensajes. Con el cifrado del lado del servidor (SSE), se proporciona cifrado de datos en reposo. Amazon SQS cifra sus datos en el nivel de mensaje cuando los almacena y descifra los mensajes para usted cuando accede a ellos. El SSE utiliza claves gestionadas en AWS Key Management Service. Siempre que autentique su solicitud y tenga permiso de acceso, no existe diferencia alguna entre obtener acceso a las colas cifradas y sin cifrar.

Para obtener más información, consulte Cifrado inactivo en Amazon SQS y Administración de claves de Amazon SQS.

Aplicación del cifrado de los datos en tránsito

Sin HTTPS (TLS), un atacante basado en la red puede espiar el tráfico de la red o manipularlo mediante un ataque como. man-in-the-middle Permitir solo las conexiones cifradas a través de HTTPS (TLS) mediante la condición `aws:SecureTransport` en la política de colas para forzar que las solicitudes utilicen SSL.

Consideración del uso de puntos de conexión de VPC para obtener acceso a Amazon SQS

Si tiene colas con las que debe poder interactuar pero que no deben estar expuestas a Internet, utilice los puntos de enlace de la VPC para poner en la cola el acceso solo a los hosts dentro de una VPC concreta. Puede utilizar las políticas de colas para controlar el acceso a las colas desde determinados puntos de conexión de Amazon VPC o desde determinadas VPC.

Los puntos de conexión de VPC de Amazon SQS brindan dos maneras de controlar el acceso a los mensajes:

- Puede controlar qué solicitudes, usuarios o grupos obtienen acceso a través de un punto de conexión de la VPC específico.
- Puede controlar qué VPC o puntos de enlace de la VPC tienen acceso a su cola con una política de colas.

Para obtener más información, consulte Puntos de conexión de Amazon Virtual Private Cloud para Amazon SQS y Creación de una política de punto de conexión de VPC para Amazon SQS.

Recursos de Amazon SQS relacionados

En la tabla siguiente se enumeran todos los recursos relacionados que podrían resultarle útiles cuando trabaje con este servicio.

Recurso	Descripción
Referencia de la API de Amazon Simple Queue Service	Descripciones de las acciones, parámetros y tipos de datos de la , así como una lista de errores que el servicio devuelve.
Amazon SQS en la Referencia de los comandos de la AWS CLI	Descripciones de los AWS CLI comandos que puede utilizar para trabajar con colas.
Regiones y puntos de enlace	Información acerca de las regiones y los puntos de conexión de Amazon SQS
Página del producto	Página web principal con información acerca de Amazon SQS.
Foro de debate	Foro de la comunidad donde los desarrolladores tratan aspectos técnicos relacionados con Amazon SQS.
AWS Información de soporte premium	La página web principal para obtener información sobre AWS Premium Support, un canal de soporte personali zado y de respuesta rápida que le ayuda a crear y ejecutar aplicaciones en AWS servicios de infraestr uctura.

Historial de documentación

En la siguiente tabla se describen los cambios importantes realizados en la Guía para desarrolladores de Amazon Simple Queue Service desde enero de 2019. Para recibir notificaciones sobre las actualizaciones de esta documentación, suscríbase a la fuente RSS.

A veces, las funciones del servicio se implementan de forma gradual en AWS las regiones en las que el servicio está disponible. Actualizamos esta documentación solo para la primera versión. No proporcionamos información sobre la disponibilidad en regiones ni anunciamos lanzamientos posteriores en regiones. Para obtener información sobre la disponibilidad regional de las funciones del servicio y suscribirse a las notificaciones sobre actualizaciones, consulte ¿Qué hay de nuevo? AWS .

Cambio	Descripción	Fecha
AWS Protocolo JSON	Realice solicitudes de API mediante el protocolo AWS JSON.	27 de julio de 2023
Nueva sección que describe las políticas AWS gestionadas para Amazon SQS y las actualizaciones de estas políticas	Amazon SQS ha agregado una nueva acción que permite enumerar las tareas de movimiento de mensajes más recientes (hasta diez) en una cola de origen específica. Esta acción está asociada a la operación de la API `ListMessageMoveTasks` .	7 de junio de 2023
Redireccionamiento de la cola de mensajes fallidos mediante las API	Configure el redirecci onamiento de la cola de mensajes fallidos utilizando las API de Amazon SQS.	7 de junio de 2023
ABAC para Amazon SQS	Control de acceso basado en atributos (ABAC) mediante etiquetas de cola para obtener	10 de noviembre de 2022

	permisos de acceso flexibles y escalables.	
Aumentos del límite de alto rendimiento de FIFO	Aumento de las cuotas predeterminadas para el modo de alto rendimiento FIFO en regiones comerciales, además de la optimización de documentos de alto rendimien to FIFO.	20 de octubre de 2022
El cifrado del servidor (SSE) está disponible de forma predeterminada	El cifrado del servidor (SSE) utiliza el cifrado propiedad de SQS (SSE-SQS) de forma predeterminada.	26 de septiembre de 2022
Está disponible la compatibi lidad con la protección de suplentes confusos de Amazon SQS	La protección del suplente confuso permite especificar nuevos encabezados en sus solicitudes, que se comprueba n con las condiciones de la política de KMS cuando se utiliza el SSE administrado por Amazon SQS.	29 de diciembre de 2021
SSE administrado está disponible	SSE administrado por Amazon SQS (SSE-SQS) es un cifrado del servidor administrado que utiliza claves de cifrado propiedad de Amazon SQS para proteger los datos confidenciales enviados a través de las colas de mensajes.	23 de noviembre de 2021

Está disponible el redirecci onamiento de cola de mensajes fallidos	Amazon SQS admite el redireccionamiento de cola de mensajes fallidos para colas estándar.	10 de noviembre de 2021
Está disponible el alto rendimiento para mensajes en colas FIFO	El alto rendimiento de las colas FIFO de Amazon SQS proporciona un mayor número de transacciones por segundo (TPS) para los mensajes en colas FIFO. Para obtener información sobre las cuotas de rendimiento, consulte Cuotas relacionadas con los mensajes.	27 de mayo de 2021
Está disponible el alto rendimiento para mensajes en colas FIFO en versión preliminar	El alto rendimiento de las colas FIFO de Amazon SQS se encuentra en versión preliminar y está sujeto a cambios. Esta característica proporciona un mayor número de transacciones por segundo (TPS) para los mensajes en colas FIFO. Para obtener información sobre las cuotas de rendimiento, consulte Cuotas relacionadas con los mensajes.	17 de diciembre de 2020
Nuevo diseño de la consola de Amazon SQS	Para simplificar los flujos de trabajo de desarrollo y producción, la consola de Amazon SQS cuenta con una nueva experiencia de usuario.	8 de julio de 2020

Amazon SQS admite la paginación para ListQueues y listDeadLetter SourceQueues	Puedes especificar el número máximo de resultados que se devolverán de una solicitud de ListQueues o lista. DeadLetter SourceQueues	22 de junio de 2020
Amazon SQS admite métricas de CloudWatch Amazon de 1 minuto en AWS todas las regiones, excepto en las regiones (EE. AWS GovCloud UU.)	La CloudWatch métrica de un minuto para Amazon SQS está disponible en todas las regiones, excepto AWS GovCloud (US) en las regiones.	9 de enero de 2020
Amazon SQS admite métricas de 1 minuto CloudWatch	La CloudWatch métrica de un minuto para Amazon SQS solo está disponible actualmente en las siguientes regiones: EE. UU. Este (Ohio), Europa (Irlanda), Europa (Estocolmo) y Asia Pacífico (Tokio).	25 de noviembre de 2019
AWS Lambda están disponibles activadores para las colas FIFO de Amazon SQS	Puede configurar los mensajes que llegan a una cola FIFO como desencadenadores de una función de Lambda.	25 de noviembre de 2019
El cifrado del servidor (SSE) para Amazon SQS está disponible en las regiones de China	SSE para Amazon SQS está disponible en las regiones de China.	13 de noviembre de 2019
Las colas FIFO están disponibles en la región de Medio Oriente (Baréin)	Las colas FIFO están disponibles en la región de Medio Oriente (Baréin).	10 de octubre de 2019

Los puntos de enlace de Amazon Virtual Private Cloud (Amazon VPC) para Amazon SQS están disponibles en las regiones (EE. UU. Este) y AWS GovCloud (EE. UU. Oeste) AWS GovCloud	Puede enviar mensajes a sus colas de Amazon SQS desde Amazon VPC en las regiones (EE. UU. Este) y AWS GovCloud (EE. UU. Oeste). AWS GovCloud	5 de septiembre de 2019
Amazon SQS permite solucionar problemas de colas mediante el uso de atributos del sistema de AWS X-Ray mensajes	Puede solucionar los problemas de los mensajes que pasan a través de las colas de Amazon SQS utilizando X-Ray. Esta versión agrega el parámetro de solicitud `MessageSystemAttribute` (que le permite enviar encabezados de rastreo X-Ray a través de Amazon SQS) a las operaciones de API `SendMessage` y `SendMessageBatch`, el atributo `AWSTraceHeader` a la operación de API `ReceiveMessage` y el tipo de datos `MessageSystemAttributeValue`.	28 de agosto de 2019

Puede etiquetar las colas de Amazon SQS en el momento de su creación	Puede utilizar una sola llamada a la API de Amazon SQS, una función AWS del SDK o un comando AWS Command Line Interface (AWS CLI) para crear una cola y especificar sus etiquetas de forma simultánea. Además, Amazon SQS admite las claves `aws:TagKeys` y `aws:RequestTag` AWS Identity and Access Management (IAM).	22 de agosto de 2019
El cliente de colas temporales de Amazon SQS ya está disponible	Las colas temporales le ayudan a ahorrar tiempo de desarrollo y costes de implementación al utilizar patrones de mensajes comunes como, por ejemplo, solicitud-respuesta. Puede utilizar el Cliente de colas temporales para crear colas temporales de alto rendimiento, rentables y administradas por la aplicación.	25 de julio de 2019
SSE para Amazon SQS está disponible en la región AWS GovCloud (EE. UU. Este)	El cifrado del lado del servidor (SSE) para Amazon SQS está disponible en AWS GovCloud la región (EE.UU. Este).	20 de junio de 2019

Las colas FIFO están disponibles en las regiones de Asia Pacífico (Hong Kong), China (Pekín), (EE. UU. Este) y AWS GovCloud (EE. UU. Oeste) AWS GovCloud	Las colas FIFO están disponibles en las regiones de Asia Pacífico (Hong Kong), China (Pekín), (EE. UU. Este) y AWS GovCloud (EE. UU. Oeste). AWS GovCloud	15 de mayo de 2019
Las políticas de punto de conexión de Amazon VPC están disponibles para Amazon SQS	Puede crear políticas de punto de conexión de Amazon VPC para Amazon SQS.	4 de abril de 2019
Las colas FIFO están disponibles en las regiones de Europa (Estocolmo) y China (Ningxia)	Las colas FIFO están disponibles en las regiones de Europa (Estocolmo) y China (Ningxia).	14 de marzo de 2019
Las colas FIFO están disponibles en todas las regiones en las que está disponible Amazon SQS	Las colas FIFO están disponibles en las regiones de Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Norte de California), Oeste de EE. UU. (Oregón), Asia-Pacífico (Bombay), Asia-Pacífico (Seúl), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney), Asia-Pacífico (Tokio), Canadá (centro), Europa (Fráncfort), Europa (Irlanda), Europa (Londres), Europa (París) y América del Sur (São Paulo).	7 de febrerode 2019

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.