aws

Guia do Desenvolvedor

Amazon Simple Queue Service

Amazon Simple Queue Service: Guia do Desenvolvedor

Table of Contents

O que é o Amazon Simple Queue Service

O Amazon Simple Queue Service (Amazon SQS) oferece uma fila hospedada segura, durável e disponível que permite integrar e desacoplar sistemas de software e componentes distribuídos. O Amazon SQS oferece constructos comuns, como filas de mensagens mortas e tags de alocação de custos. Ele fornece uma API genérica de serviços da Web que você pode acessar usando qualquer linguagem de programação compatível com o AWS SDK.

Tópicos

Benefícios do uso do Amazon SQS

- Segurança: você controla quem pode enviar e receber mensagens em uma fila do Amazon SQS. Você pode optar pela transmissão de dados sigilosos protegendo o conteúdo das mensagens em filas por meio da criptografia do lado do servidor (SSE) gerenciada pelo Amazon SQS ou das chaves de SSE gerenciadas no AWS Key Management Service (AWS KMS).
- Durabilidade: para garantir a segurança de suas mensagens, o Amazon SQS as armazena em vários servidores. As filas padrão oferecem suporte à entrega de at-least-once mensagens, e as filas FIFO oferecem suporte ao processamento de mensagens exatamente uma vez e ao modo de alto rendimento.
- Disponibilidade: o Amazon SQS usa infraestrutura redundante para fornecer acesso altamente simultâneo às mensagens, além de alta disponibilidade para produzir e consumir mensagens.
- Escalabilidade: o Amazon SQS pode processar cada solicitação em buffer de forma independente, escalando de forma transparente para lidar com qualquer aumento ou pico de carga sem nenhuma instrução de provisionamento.
- Confiabilidade: o Amazon SQS bloqueia suas mensagens durante o processamento, para que vários produtores possam enviar, e vários consumidores possam receber, mensagens ao mesmo tempo.
- Personalização: suas filas não precisam ser exatamente iguais. Você pode definir um atraso padrão em uma fila, por exemplo. Você pode armazenar o conteúdo de mensagens maiores que

256 KB usando o Amazon Simple Storage Service (Amazon S3) ou o Amazon DynamoDB, com o Amazon SQS mantendo um ponteiro no objeto do Amazon S3, ou pode dividir uma mensagem grande em mensagens menores.

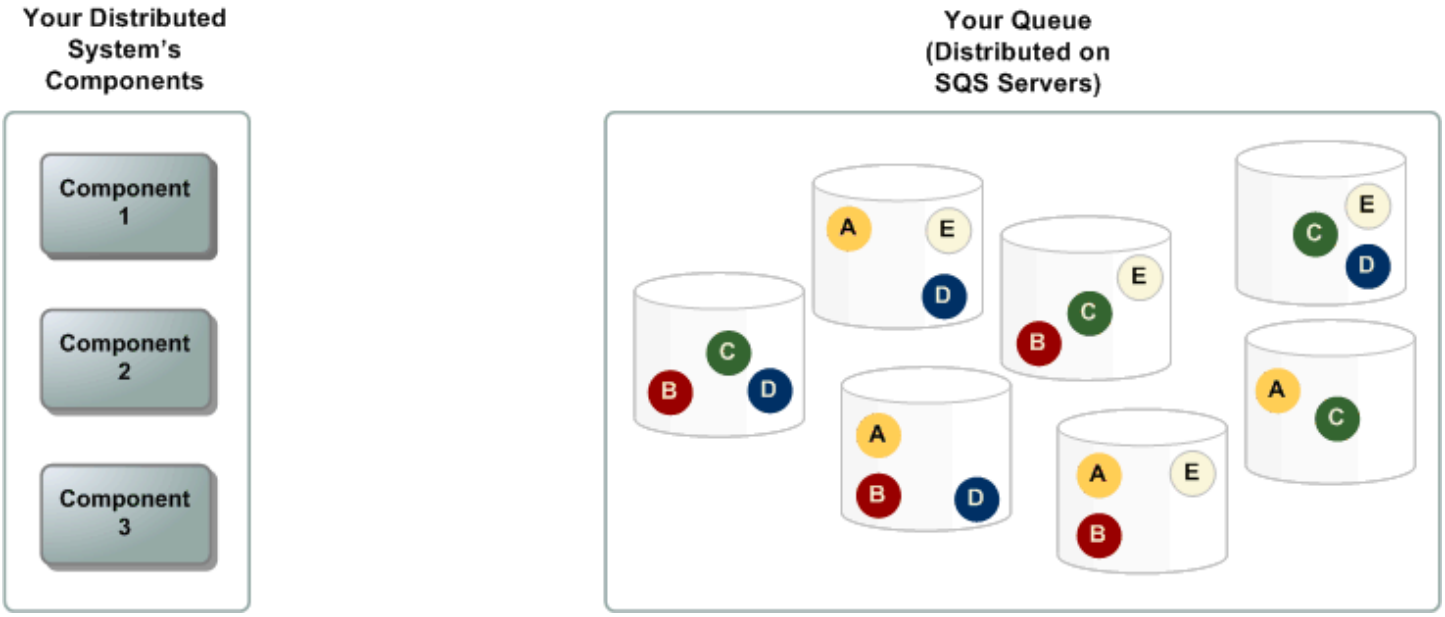
Arquitetura básica do Amazon SQS

Esta seção descreve as partes de um sistema de mensagens distribuído e explica o ciclo de vida de uma mensagem do Amazon SQS.

Filas distribuídas

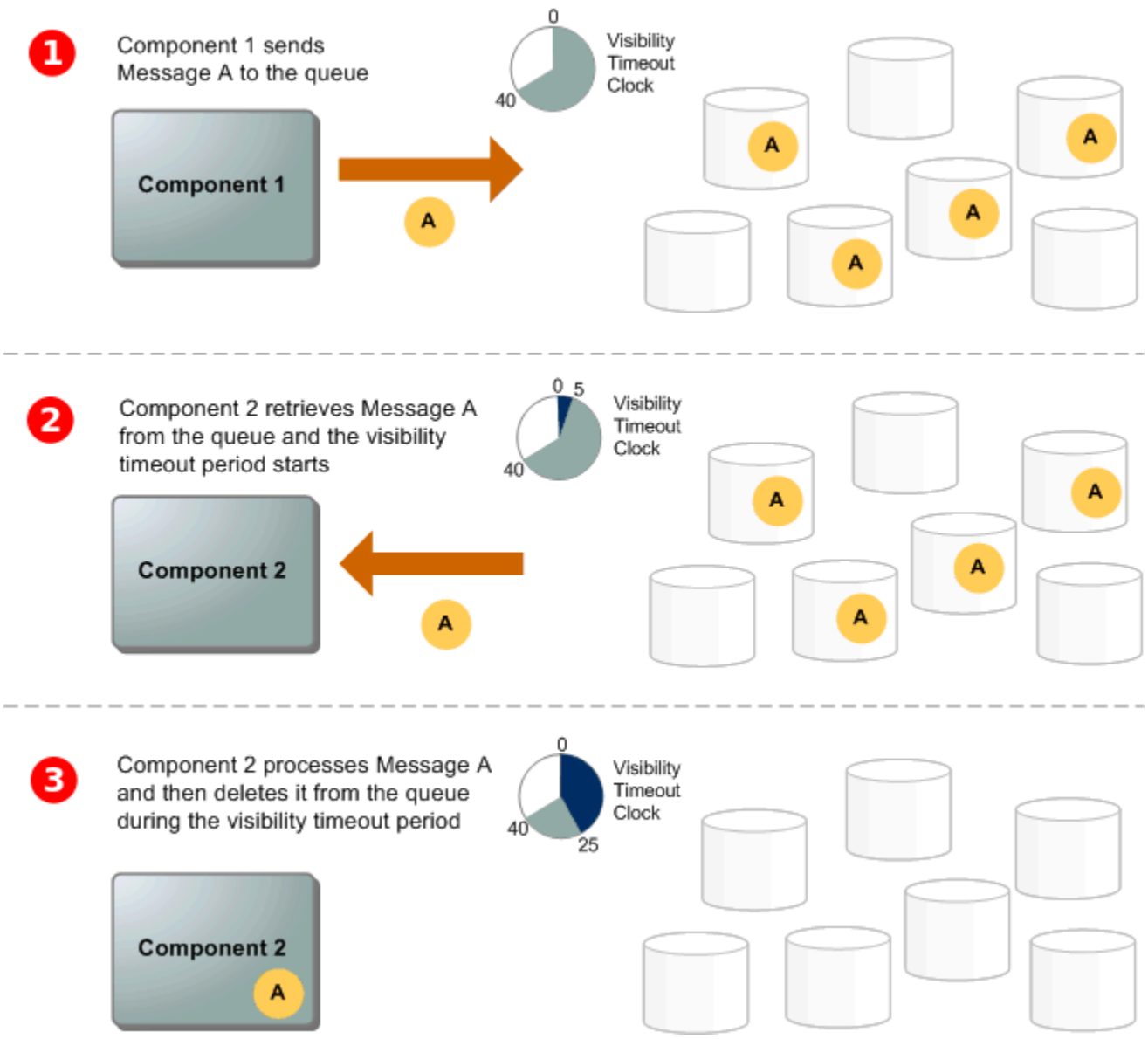
Há três partes principais em um sistema de mensagens distribuído: os componentes do sistema distribuído, a fila (distribuída em servidores do Amazon SQS) e as mensagens na fila.

No cenário a seguir, o sistema tem vários produtores (componentes que enviam mensagens para a fila) e consumidores (componentes que recebem mensagens da fila). A fila (que contém as mensagens A a E) armazena as mensagens de forma redundante em vários servidores do Amazon SQS.



Ciclo de vida de mensagens

O cenário a seguir descreve o ciclo de vida de uma mensagem do Amazon SQS em uma fila, da criação à exclusão.



1

Um produtor (componente 1) envia a mensagem A para uma fila, e a mensagem é distribuída pelos servidores do Amazon SQS de forma redundante.

2

Quando um consumidor (componente 2) está pronto para processar mensagens, ele consume as mensagens da fila, e a mensagem A é devolvida. Enquanto a mensagem A está sendo processada, ela permanece na fila e não é devolvida para as solicitações de recebimento subsequentes durante todo o tempo limite de visibilidade.

3

O consumidor (componente 2) exclui a mensagem A da fila para impedir que a mensagem seja recebida e processada novamente quando o tempo limite de visibilidade for esgotado.

> **Note**
>
> O Amazon SQS exclui automaticamente as mensagens que estiverem em uma fila por mais tempo que o período de retenção máximo de mensagens. O período de retenção de mensagens padrão é de quatro dias. No entanto, você pode configurar o período de retenção de mensagens em um valor de 60 segundos a 1.209.600 segundos (14 dias) usando a ação `SetQueueAttributes`

Diferenças entre o Amazon SQS, Amazon MQ e Amazon SNS

O Amazon SQS, o Amazon SNS e o Amazon MQ oferecem serviços de mensagens altamente escaláveis easy-to-use e gerenciados, cada um projetado para funções específicas em sistemas distribuídos. Aqui está uma visão geral aprimorada das diferenças entre esses serviços:

O Amazon SQS separa e escala sistemas e componentes de software distribuídos como um serviço de fila. Normalmente, ele processa mensagens por meio de um único assinante, ideal para fluxos de trabalho em que a prevenção de pedidos e perdas é fundamental. Para uma distribuição mais ampla, a integração do Amazon SQS com o Amazon SNS permite um padrão de mensagens de fanout, enviando mensagens de forma eficaz para vários assinantes ao mesmo tempo.

O Amazon SNS permite que os editores enviem mensagens para vários assinantes por meio de tópicos, que servem como canais de comunicação. Os assinantes recebem mensagens publicadas usando um tipo de endpoint compatível Amazon Data Firehose, como Amazon SQS, Lambda, HTTP, e-mail, notificações push móveis e mensagens de texto móveis (SMS). Esse serviço é ideal para cenários que exigem notificações imediatas, como engajamento do usuário em tempo real ou sistemas de alarme. Para evitar a perda de mensagens quando os assinantes estão off-line, a integração do Amazon SNS com as mensagens de fila do Amazon SQS garante uma entrega consistente.

O Amazon MQ se adapta melhor às empresas que desejam migrar de agentes de mensagens tradicionais, oferecendo suporte a protocolos de mensagens padrão, como AMQP e MQTT, junto com o Apache ActiveMQ e o RabbitMQ. Ele oferece compatibilidade com sistemas legados que precisam de mensagens estáveis e confiáveis sem reconfiguração significativa.

O gráfico a seguir fornece uma visão geral do tipo de recurso de cada serviço:

Tipo de recurso	Amazon SNS	Amazon SQS	Amazon MQ
Síncrona	Não	Não	Sim
Assíncrona	Sim	Sim	Sim
Filas	Não	Sim	Sim
Sistema de publicador e assinante de mensagens	Sim	Não	Sim
Agente de mensagens	Não	Não	Sim

O Amazon SQS e o Amazon SNS são recomendamos para novas aplicações que podem se beneficiar de uma escalabilidade praticamente ilimitada e de APIs simples. Eles geralmente oferecem soluções mais econômicas para aplicações de alto volume com seus preços. pay-as-you-go Recomendamos o Amazon MQ para migrar aplicativos de agentes de mensagens existentes que dependem da compatibilidade com APIs como JMS ou protocolos como Advanced Message Queuing Protocol (AMQP), MQTT e Simple Text Oriented Message Protocol (STOMP). OpenWire

Configurar o Amazon SQS

Antes de usar o Amazon SQS pela primeira vez, siga as etapas abaixo.

Tópicos

Etapa 1: criar um usuário Conta da AWS e IAM

Para acessar qualquer AWS serviço, primeiro você precisa criar uma Conta da AWSconta da Amazon.com que possa usar AWS produtos. Você pode usar o seu Conta da AWS para visualizar seus relatórios de atividade e uso e para gerenciar a autenticação e o acesso.

Para evitar o uso do usuário Conta da AWS raiz para ações do Amazon SQS, é uma prática recomendada criar um usuário do IAM para cada pessoa que precisa de acesso administrativo ao Amazon SQS.

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções on-line.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Signing in as the root user (Fazer login como usuário-raiz) no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

   Para obter instruções, consulte Habilitar AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Create a permission set no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Add groups no Guia do usuário do AWS IAM Identity Center .

Etapa 2: conceder acesso programático

Para usar ações do Amazon SQS (por exemplo, usando Java ou por meio do AWS Command Line Interface), você precisa de um ID de chave de acesso e uma chave de acesso secreta.

> **Note**
>
> O ID da chave de acesso e a chave de acesso secreta são específicos de AWS Identity and Access Management. Não os confunda com credenciais de outros AWS serviços, como pares de chaves do Amazon EC2.

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do AWS Management Console. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
Identificação da força de trabalho	Use credenciais temporárias para assinar solicitações	Siga as instruções da interface que deseja utilizar.

Qual usuário precisa de acesso programático?	Para	Por
(Usuários gerenciados no Centro de Identidade do IAM)	programáticas para AWS SDKs ou APIs. AWS CLI AWS	• Para o AWS CLI, consulte Configurando o AWS CLI para uso AWS IAM Identity Center no Guia do AWS Command Line Interface usuário. • Para AWS SDKs, ferrament as e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de AWS SDKs e ferramentas.
IAM	Use credenciais temporári as para assinar solicitações programáticas para AWS SDKs ou APIs. AWS CLI AWS	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitaç ões programáticas para AWS SDKs AWS CLI ou APIs. AWS	Siga as instruções da interface que deseja utilizar. • Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. • Para AWS SDKs e ferramentas, consulte Autenticar usando credencia is de longo prazo no Guia de referência de AWS SDKs e ferramentas. • Para AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Etapa 3: preparar-se para usar o código de exemplo

Este guia inclui exemplos que usam o AWS SDK for Java. Para executar o código de exemplo, siga as instruções de configuração em Conceitos básicos do AWS SDK for Java 2.0.

Você pode desenvolver AWS aplicativos em outras linguagens de programação, como GoJavaScript, Python e Ruby. Para obter mais informações, consulte Ferramentas para desenvolver AWS.

> **Note**
>
> Você pode explorar o Amazon SQS sem escrever código com ferramentas como o AWS Command Line Interface (AWS CLI) ou o Windows. PowerShell Você pode encontrar AWS CLI exemplos na seção Amazon SQS da Referência de AWS CLI Comandos. Você pode

encontrar PowerShell exemplos do Windows na seção Amazon Simple Queue Service da Referência de AWS Tools for PowerShell Cmdlet.

Próximas etapas

Agora, você já está com tudo pronto para começar a gerenciar filas e mensagens do Amazon SQS usando o AWS Management Console.

Conceitos básicos do Amazon SQS

Nesta seção, você aprenderá a criar filas padrão ou FIFO usando o console do Amazon SQS.

Tópicos

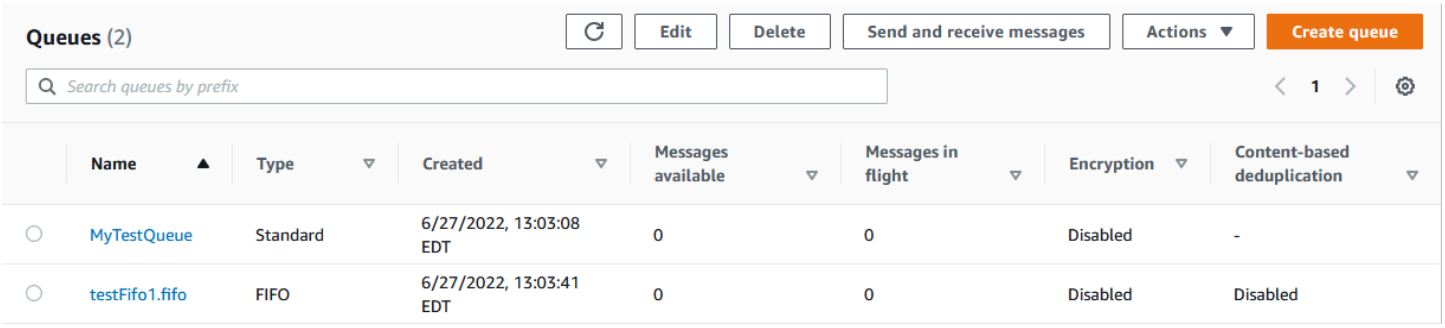
Pré-requisitos

Antes de começar, conclua as etapas em Configurar o Amazon SQS.

Noções básicas sobre o console do Amazon SQS

Ao abrir o console do Amazon SQS, escolha Filas no painel de navegação. A página Queues (Filas) fornece informações sobre todas as filas na região ativa.

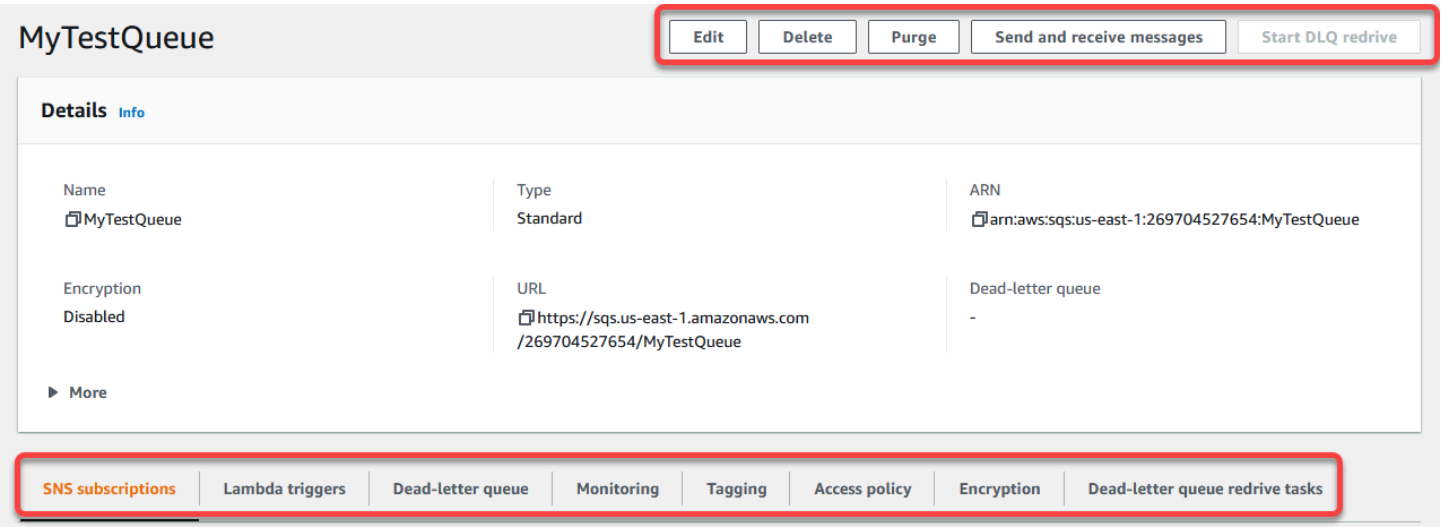
Cada entrada da fila fornece informações essenciais sobre a fila, incluindo seu tipo e atributos principais. As filas padrão, otimizadas para máxima taxa de transferência e melhor ordenação de mensagens, são diferenciadas das filas First-In-First-Out (FIFO), que priorizam a ordenação e a exclusividade das mensagens para aplicativos que exigem um sequenciamento estrito de mensagens.



Elementos e ações interativos

Na página Filas, você tem várias opções para gerenciar suas filas:

1. Ações rápidas — Ao lado de cada nome de fila, um menu suspenso oferece acesso rápido a ações comuns, como enviar mensagens, visualizar ou excluir mensagens, configurar gatilhos e excluir a própria fila.
2. Visualização e configuração detalhadas — Clicar no nome de uma fila abre sua página de detalhes, onde você pode se aprofundar nas configurações e configurações da fila. Aqui, você pode ajustar parâmetros como período de retenção de mensagens, tempo limite de visibilidade e tamanho máximo da mensagem para adaptar a fila aos requisitos do seu aplicativo.



Seleção de região e tags de recursos

Certifique-se de que você está no caminho certo Região da AWS para acessar e gerenciar suas filas de forma eficaz. Além disso, considere a utilização de tags de recursos para organizar e categorizar suas filas, permitindo melhor gerenciamento de recursos, alocação de custos e controle de acesso em seu ambiente compartilhado. AWS

Ao aproveitar os recursos e funcionalidades oferecidos no console do Amazon SQS, você pode gerenciar com eficiência sua infraestrutura de mensagens, otimizar o desempenho das filas e garantir a entrega confiável de mensagens para seus aplicativos.

Tipos de fila do Amazon SQS

O Amazon SQS é compatível com dois tipos de fila: filas comuns e filas FIFO. Use as informações da tabela a seguir para escolher a fila certa para sua situação. Para saber mais sobre as filas do Amazon SQS, consulte Conceitos básicos de filas padrão do Amazon SQS e Introdução às filas FIFO no Amazon SQS.

Filas padrão	Filas FIFO
Throughput ilimitado: as filas padrão oferecem suporte a um número quase ilimitado de chamadas de API por segundo, por ação de API (`SendMessage` , `ReceiveMessage` ou `DeleteMessage`). Entrega pelo menos uma vez: uma mensagem é entregue pelo menos uma vez, mas às vezes, mais de uma cópia da mensagem é entregue. Melhor ordenação possível: às vezes, as mensagens podem ser entregues em uma ordem diferente da qual elas foram enviadas.	Alta taxa de transferência: se você usar o processamento em lotes, as filas FIFO oferecem suporte a até 3.000 mensagens por segundo, por método de API (`SendMessageBatch` , `ReceiveMessage` ou `DeleteMessageBatch`). As 3 mil mensagens por segundo representam 300 chamadas de API, cada uma com um lote de 10 mensagens. Para solicitar um aumento, envie um pedido de suporte. Sem o agrupamento em lote, as filas FIFO oferecem suporte a até 300 chamadas de API por segundo, por método de API (`SendMessage` , `ReceiveMessage` ou `DeleteMessage`). Processamento exatamente uma vez: uma mensagem é entregue uma vez e permanece disponível até que um consumidor a processe e exclua. As duplicações não são introduzidas na fila. Entrega primeiro a entrar, primeiro a sair: a ordem em que as mensagens são enviadas e recebidas é estritamente preservada.
	
Enviar dados entre aplicativos quando a taxa de transferência for importante, por exemplo: • Desacoplar solicitações de usuário em tempo real do intenso trabalho em segundo plano: permite que os usuários façam upload	Enviar dados entre aplicativos quando a ordem dos eventos for importante, por exemplo: • Garantir que os comandos inseridos pelo usuário sejam executados na ordem correta.

Filas padrão	Filas FIFO
de mídia enquanto a redimensionam ou a codificam. • Alocar tarefas para vários nós de processam ento: processa um número elevado de solicitações de validação de cartão de crédito. • Organizar as mensagens em lote para processamento futuro: programa várias entradas para adicioná-las ao banco de dados.	• Exibir o preço do produto correto enviando modificações de preço na ordem correta. • Impedir que um aluno se inscreva em um curso antes de criar uma conta.

Criar uma fila padrão do Amazon SQS e enviar uma mensagem

Veja como criar uma fila padrão para o Amazon SQS.

Crie uma fila usando o console do Amazon SQS

É possível usar o console do Amazon SQS para criar filas padrão. O console fornece valores padrão para todas as configurações, exceto para o nome da fila.

> **Important**
>
> Em 17 de agosto de 2022, a criptografia do lado do servidor (SSE) padrão foi aplicada a todas as filas do Amazon SQS.
> Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de filas. Os nomes das filas podem ser acessados por muitos Amazon Web Services, incluindo faturamento e CloudWatch registros. Os nomes de filas não devem ser usados para dados privados ou sigilosos.

Para criar uma fila padrão do Amazon SQS

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. Selecione Criar fila.
3. Para o tipo, a fila do tipo padrão é definida por padrão.

> **Note**
>
> Não é possível alterar o tipo de uma fila depois de criá-la.

4. Insira um Name (Nome) para a fila.
5. (Opcional) O console define valores padrão para os parâmetros de configuração da fila. Em Configuration (Configuração), você pode definir novos valores para os seguintes parâmetros:

   a. Em Visibility timeout (Tempo limite de visibilidade), insira a duração e as unidades. O intervalo é de 0 segundo a 12 horas. O valor padrão de é 30 segundos.

   b. Em Message retention period (Período de retenção de mensagens), insira a duração e as unidades. O intervalo é de 1 minuto a 14 dias. O valor padrão é 4 dias.

   c. Em Delivery delay (Atraso de entrega), insira a duração e as unidades. O intervalo é de 0 segundo a 15 minutos. O valor de padrão é 0 segundos.

   d. Em Maximum message size (Tamanho máximo da mensagem), insira um valor. O intervalo é de 1 KB a 256 KB. O valor padrão é 256 KB.

   e. Em Receive message wait time (Tempo de espera da mensagem), insira um valor. O intervalo é de 0 a 20 segundos. O valor padrão é 0 segundo, o que define uma sondagem curta. Qualquer valor diferente de zero define uma sondagem longa.

6. (Opcional) Defina uma política de acesso. A política de acesso define as contas, usuários e funções que podem acessar a fila. A política de acesso também define as ações (como `SendMessage`, `ReceiveMessage` ou `DeleteMessage`) que os usuários podem acessar. A política padrão permite que apenas o proprietário da fila envie e receba mensagens.

   Para definir a política de acesso, realize um dos seguintes procedimentos:

   - Escolha Basic (Básico) para configurar quem pode enviar mensagens para a fila e quem pode receber mensagens dela. O console cria a política com base em suas escolhas e exibe a política de acesso resultante no painel JSON somente leitura.
   - Escolha Advanced (Avançado) para modificar a política de acesso JSON diretamente. Isso permite que você especifique um conjunto personalizado de ações que cada entidade (conta, usuário ou função) pode executar.

7. Em Redrive allow policy (Política de permissão de redirecionamento), escolha Enabled (Habilitada). Selecione uma das seguintes opções: Allow all (Permitir tudo), By queue (Por fila)

ou Deny all (Negar tudo). Ao escolher By queue (Por fila), especifique uma lista de até 10 filas de origem pelo nome do recurso da Amazon (ARN).

8. O Amazon SQS fornece criptografia do lado do servidor gerenciada por padrão. Para escolher um tipo de chave de criptografia ou desabilitar a criptografia do lado do servidor gerenciada pelo Amazon SQS, expanda Encryption (Criptografia). Para obter mais informações sobre os tipos de chave de criptografia, consulte Configurando a criptografia do lado do servidor para uma fila usando chaves de criptografia gerenciadas pelo SQS e Configurando a criptografia do lado do servidor para uma fila usando o console do Amazon SQS.

   > **Note**
   >
   > Com a SSE habilitada, as solicitações anônimas `SendMessage` e `ReceiveMessage` à fila criptografada serão rejeitadas. As práticas recomendadas de segurança do Amazon SQS não aconselham o uso de solicitações anônimas. Se você quiser enviar solicitações anônimas a uma fila do Amazon SQS, desabilite o SSE.

9. (Opcional) Para configurar uma fila de mensagens mortas para receber mensagens que não podem ser entregues, expanda Dead-letter queue (Fila de mensagens mortas).
10. (Opcional) Para adicionar tags à fila, expanda Tags.
11. Selecione Criar fila. O Amazon SQS cria a fila e exibe a página Details (Detalhes) da fila.

O Amazon SQS propaga as informações sobre a nova fila pelo sistema. Como o Amazon SQS é um sistema distribuído, você pode enfrentar um pequeno atraso antes que o console exiba a fila na página Queues (Filas).

Enviar uma mensagem

Depois de criar sua fila, você pode enviar uma mensagem para ela.

1. No painel de navegação à esquerda, escolha Queues (Filas). Na lista de filas, selecione a fila que você criou.
2. Em Actions (Ações), escolha Send and receive messages (Enviar e receber mensagens).

   O console exibe a página Send and receive messages (Enviar e receber mensagens).
3. Em Message (Mensagem), insira o texto da mensagem.

4. Para uma fila padrão, é possível inserir um valor para Atraso de entrega e escolher as unidades. Por exemplo, insira 60 e escolha seconds (segundos). Para ter mais informações, consulte Temporizadores de mensagens do Amazon SQS.
5. Escolha Send Message (Enviar mensagem).

   Quando a mensagem é enviada, o console exibe uma mensagem de sucesso. Escolha View details (Visualizar os detalhes) para exibir informações sobre a mensagem enviada.

Criar uma fila FIFO do Amazon SQS e enviar uma mensagem

Veja como criar uma fila FIFO para o Amazon SQS.

Criar uma fila

É possível usar o console do Amazon SQS para criar filas FIFO. O console fornece valores padrão para todas as configurações, exceto para o nome da fila.

> **Important**
>
> Em 17 de agosto de 2022, a criptografia do lado do servidor (SSE) padrão foi aplicada a todas as filas do Amazon SQS.
> Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de filas. Os nomes das filas podem ser acessados por muitos Amazon Web Services, incluindo faturamento e CloudWatch registros. Os nomes de filas não devem ser usados para dados privados ou sigilosos.

Para criar uma fila FIFO do Amazon SQS

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. Selecione Criar fila.
3. Para o tipo, a fila do tipo padrão é definida por padrão. Para criar uma fila FIFO, escolha FIFO.

   > **Note**
   >
   > Não é possível alterar o tipo de uma fila depois de criá-la.

4. Insira um Name (Nome) para a fila.

   O nome de uma fila FIFO deve terminar com o sufixo `.fifo`. O sufixo conta para a cota de 80 caracteres do nome da fila. Para determinar se uma fila é FIFO, você pode conferir se o nome da fila termina com o sufixo.

5. (Opcional) O console define valores padrão para os parâmetros de configuração da fila. Em Configuration (Configuração), você pode definir novos valores para os seguintes parâmetros:

   a. Em Visibility timeout (Tempo limite de visibilidade), insira a duração e as unidades. O intervalo é de 0 segundo a 12 horas. O valor padrão de é 30 segundos.

   b. Em Message retention period (Período de retenção de mensagens), insira a duração e as unidades. O intervalo é de 1 minuto a 14 dias. O valor padrão é 4 dias.

   c. Em Delivery delay (Atraso de entrega), insira a duração e as unidades. O intervalo é de 0 segundo a 15 minutos. O valor de padrão é 0 segundos.

   d. Em Maximum message size (Tamanho máximo da mensagem), insira um valor. O intervalo é de 1 KB a 256 KB. O valor padrão é 256 KB.

   e. Em Receive message wait time (Tempo de espera da mensagem), insira um valor. O intervalo é de 0 a 20 segundos. O valor padrão é 0 segundo, o que define uma sondagem curta. Qualquer valor diferente de zero define uma sondagem longa.

   f. Para uma fila FIFO, escolha Content-based deduplication (Eliminação de duplicação baseada em conteúdo) para habilitar a eliminação de duplicação baseada em conteúdo. Por padrão, essa configuração está desabilitada.

   g. (Opcional) Em uma fila FIFO, para habilitar um throughput mais alto a fim de enviar e receber mensagens na fila, escolha Enable high throughput FIFO (Habilitar FIFO de alto throughput).

      Escolher esta opção altera as opções relacionadas (Deduplication scope [Escopo de eliminação de duplicação] e FIFO throughput limit [Limite de transferência FIFO]) para as configurações necessárias a fim de habilitar a alta taxa de transferência para filas FIFO. Se você alterar qualquer uma das configurações necessárias para usar FIFO de alta taxa de transferência, a taxa de transferência normal permanecerá em vigor para a fila e a eliminação de duplicação ocorrerá conforme especificado. Para obter mais informações, consulte Alta taxa de transferência para filas FIFO no Amazon SQS e Cotas de mensagens do Amazon SQS.

6. (Opcional) Defina uma política de acesso. A política de acesso define as contas, usuários e funções que podem acessar a fila. A política de acesso também define as ações (como

`SendMessage`, `ReceiveMessage` ou `DeleteMessage`) que os usuários podem acessar. A política padrão permite que apenas o proprietário da fila envie e receba mensagens.

Para definir a política de acesso, realize um dos seguintes procedimentos:

- Escolha Basic (Básico) para configurar quem pode enviar mensagens para a fila e quem pode receber mensagens dela. O console cria a política com base em suas escolhas e exibe a política de acesso resultante no painel JSON somente leitura.
- Escolha Advanced (Avançado) para modificar a política de acesso JSON diretamente. Isso permite que você especifique um conjunto personalizado de ações que cada entidade (conta, usuário ou função) pode executar.

7. Em Redrive allow policy (Política de permissão de redirecionamento), escolha Enabled (Habilitada). Selecione uma das seguintes opções: Allow all (Permitir tudo), By queue (Por fila) ou Deny all (Negar tudo). Ao escolher By queue (Por fila), especifique uma lista de até 10 filas de origem pelo nome do recurso da Amazon (ARN).
8. O Amazon SQS fornece criptografia do lado do servidor gerenciada por padrão. Para escolher um tipo de chave de criptografia ou desabilitar a criptografia do lado do servidor gerenciada pelo Amazon SQS, expanda Encryption (Criptografia). Para obter mais informações sobre os tipos de chave de criptografia, consulte Configurando a criptografia do lado do servidor para uma fila usando chaves de criptografia gerenciadas pelo SQS e Configurando a criptografia do lado do servidor para uma fila usando o console do Amazon SQS.

   > **Note**
   >
   > Com a SSE habilitada, as solicitações anônimas `SendMessage` e `ReceiveMessage` à fila criptografada serão rejeitadas. As práticas recomendadas de segurança do Amazon SQS não aconselham o uso de solicitações anônimas. Se você quiser enviar solicitações anônimas a uma fila do Amazon SQS, desabilite o SSE.

9. (Opcional) Para configurar uma fila de mensagens mortas para receber mensagens que não podem ser entregues, expanda Dead-letter queue (Fila de mensagens mortas).
10. (Opcional) Para adicionar tags à fila, expanda Tags.
11. Selecione Criar fila. O Amazon SQS cria a fila e exibe a página Details (Detalhes) da fila.

O Amazon SQS propaga as informações sobre a nova fila pelo sistema. Como o Amazon SQS é um sistema distribuído, você pode enfrentar um pequeno atraso antes que o console exiba a fila na página Queues (Filas).

Depois de criar uma fila, você pode enviar mensagens para ela e receber e excluir mensagens. Você também pode editar qualquer uma das definições de configuração de fila, exceto o tipo de fila.

Enviar uma mensagem

Depois de criar sua fila, você pode enviar uma mensagem para ela.

1. No painel de navegação à esquerda, escolha Queues (Filas). Na lista de filas, selecione a fila que você criou.
2. Em Actions (Ações), escolha Send and receive messages (Enviar e receber mensagens).

   O console exibe a página Send and receive messages (Enviar e receber mensagens).
3. Em Message (Mensagem), insira o texto da mensagem.
4. Para uma fila FIFO (primeiro a entrar, primeiro a sair), insira um ID do grupo de mensagens. Para ter mais informações, consulte Lógica de entrega de filas FIFO no Amazon SQS.
5. (Opcional) Para uma fila FIFO, você pode inserir um ID de eliminação de duplicação de mensagens. Se você habilitou a eliminação de duplicação baseada em conteúdo para a fila, o ID de eliminação de duplicação de mensagens não será necessário. Para ter mais informações, consulte Lógica de entrega de filas FIFO no Amazon SQS.
6. As filas FIFO não são compatíveis com temporizadores em mensagens individuais. Para ter mais informações, consulte Temporizadores de mensagens do Amazon SQS.
7. Escolha Send Message (Enviar mensagem).

   Quando a mensagem é enviada, o console exibe uma mensagem de sucesso. Escolha View details (Visualizar os detalhes) para exibir informações sobre a mensagem enviada.

Gerenciar uma fila do Amazon SQS

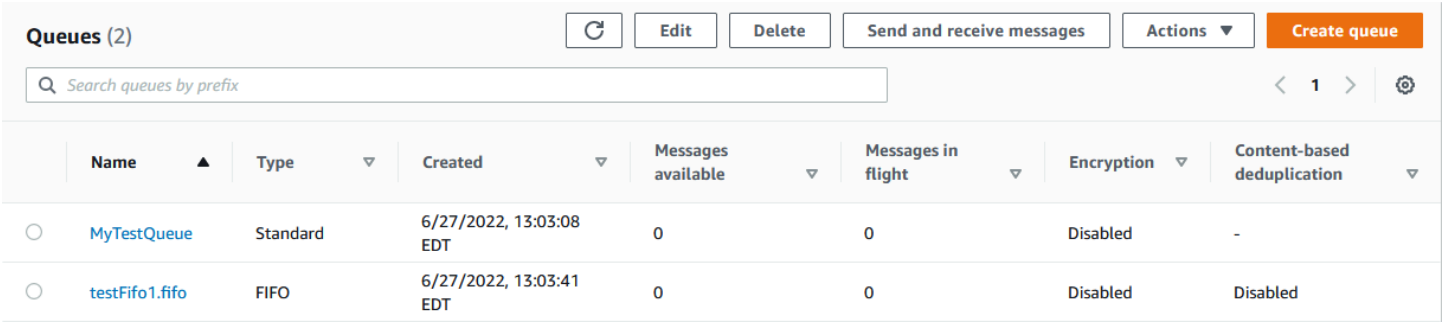
Esta seção ajudará você a se familiarizar com o Amazon SQS, mostrando como gerenciar filas e mensagens usando o console do Amazon SQS.

Pré-requisitos

Antes de começar, conclua as etapas em Configurar o Amazon SQS.

Noções básicas sobre o console do Amazon SQS

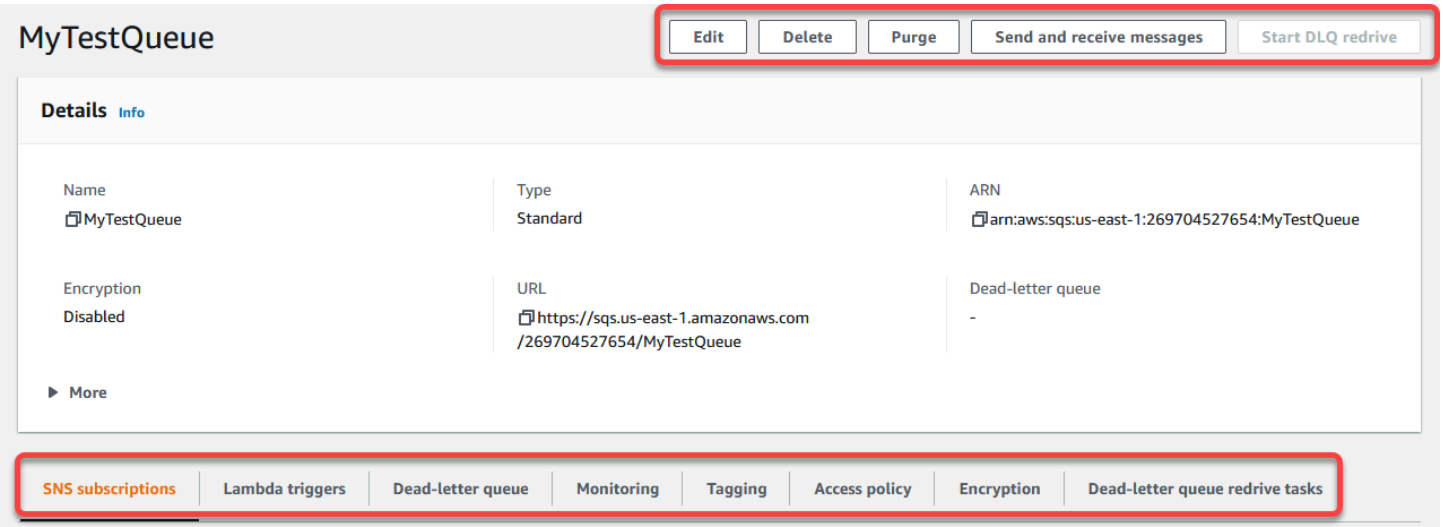
Ao abrir o console, escolha Queues (Filas) no painel de navegação para exibir a página Queues (Filas). A página Queues (Filas) fornece informações sobre todas as filas na região ativa.



A entrada de cada fila mostra o tipo de fila e outras informações sobre ela. A coluna Type (Tipo) ajuda a diferenciar rapidamente as filas padrão das filas FIFO (primeiro a entrar, primeiro a sair).

Na página Queues (Filas), há duas maneiras de executar ações em uma fila. Você pode escolher a opção ao lado do nome da fila e escolher a ação que deseja executar nela.

Você também pode escolher o nome da fila, que abre a página Details (Detalhes) da fila. A página Details (Detalhes) inclui as mesmas ações que a página Queues (Filas). Além disso, você pode escolher uma das guias abaixo da seção Details (Detalhes) para exibir detalhes e ações adicionais de configuração.



Editando uma fila do Amazon SQS usando o console

Você pode usar o console do Amazon SQS para editar quaisquer parâmetros de configuração de fila (exceto o tipo de fila) e adicionar ou remover recursos de fila.

Para editar uma fila do Amazon SQS (console)

1. Abra a página Queues (Filas) do console do Amazon SQS.
2. Selecione uma fila e escolha Edit (Editar).
3. (Opcional) Em Configuration (Configuração), atualize os parâmetros de configuração da fila.
4. (Opcional) Para atualizar a política de acesso, em Access policy (Política de acesso), modifique a política JSON.
5. (Opcional) Para atualizar a política de permissão de redirecionamento de uma fila de mensagens não entregues, expanda Redrive allow policy (Política de permissão de redirecionamento).
6. (Opcional) Para atualizar ou remover a criptografia, expanda Encryption (Criptografia).
7. (Opcional) Para adicionar, atualizar ou remover uma fila de mensagens mortas (que permite receber mensagens que não podem ser entregues), expanda Dead-letter queue (Fila de mensagens mortas).
8. (Opcional) Para adicionar, atualizar ou remover as tags da fila, expanda Tags.
9. Escolha Salvar.

   O console exibe a página Details (Detalhes) da fila.

Recebendo e excluindo uma mensagem no Amazon SQS

Depois de enviar mensagens para uma fila do Amazon SQS, você tem a opção de recebê-las e excluí-las. Ao solicitar mensagens de uma fila, você não pode especificar mensagens individuais. Em vez disso, você determina o número máximo de mensagens que deseja recuperar, até um limite de 10.

O Amazon SQS opera como um sistema distribuído, o que pode ocasionalmente resultar em uma resposta vazia ao recuperar mensagens de uma fila com poucas mensagens. Se isso acontecer, basta executar novamente sua solicitação. Para otimizar a recuperação de mensagens e minimizar respostas vazias, considere usar uma sondagem longa. A pesquisa longa atrasa a resposta até que uma mensagem fique disponível ou que a pesquisa atinja o tempo limite, reduzindo os custos desnecessários da pesquisa e melhorando a eficiência.

As mensagens não são excluídas automaticamente após a recuperação porque o Amazon SQS garante que você não perca o acesso a uma mensagem devido a falhas de processamento, como problemas com seu aplicativo ou interrupções na rede. Para remover permanentemente uma mensagem da fila, você deve enviar explicitamente uma solicitação de exclusão após o processamento da mensagem para confirmar o recebimento e o tratamento bem-sucedidos.

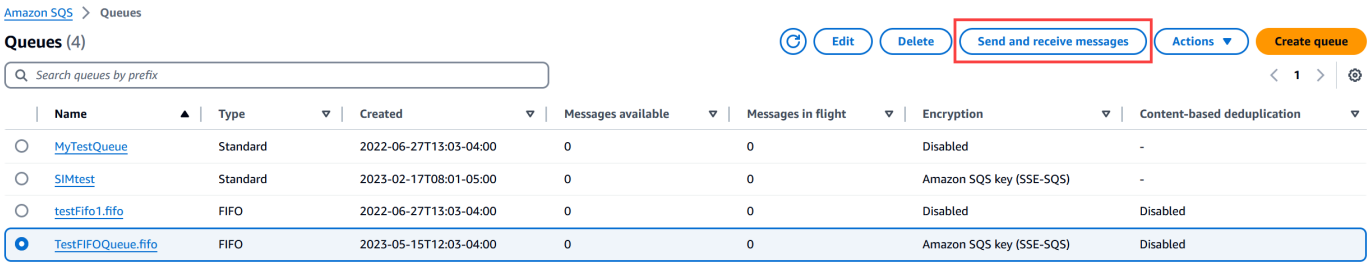
Quando as mensagens são recuperadas por meio do console do Amazon SQS, elas ficam imediatamente visíveis novamente para recuperação. Esse comportamento padrão garante que as mensagens não sejam perdidas inadvertidamente durante as operações manuais, mas podem levar ao processamento repetido. Em ambientes automatizados, ajuste a configuração do tempo limite de visibilidade para controlar por quanto tempo uma mensagem permanece invisível para outros consumidores após ser recuperada. Essa configuração é crucial para coordenar o processamento de mensagens em vários consumidores e garantir que as mensagens sejam processadas apenas uma vez.

Para operações mais detalhadas sobre recebimento e exclusão de mensagens, consulte o Guia de referência da API Amazon SQS. Este guia oferece informações abrangentes sobre endpoints de API, incluindo parâmetros que gerenciam cenários complexos de tratamento de mensagens de forma eficaz.

Para receber e excluir uma mensagem usando o console

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.

3. Na página Filas, selecione uma fila e escolha Enviar e receber mensagens.



4. Na página Enviar e receber mensagens, escolha Sondagem de mensagens.

   O Amazon SQS começa a sondar mensagens na fila. A barra de progresso no lado direito da seção Receive messages (Receber mensagens) exibe a duração da sondagem.

   A seção Messages (Mensagens) exibe uma lista das mensagens recebidas. Para cada mensagem, a lista exibe o ID da mensagem, a data de envio, o tamanho e a contagem de recebimento.

5. Para excluir mensagens, escolha as mensagens que você deseja excluir e, em seguida, escolha Excluir.
6. Na caixa de diálogo Excluir mensagens, escolha Excluir.

Confirmando que uma fila do Amazon SQS está vazia

Na maioria dos casos, você pode usar a sondagem longa para determinar se uma fila está vazia. Em casos raros, você pode receber respostas vazias mesmo quando uma fila ainda contém mensagens, especialmente se você especificar um valor baixo para o tempo de espera da mensagem quando criar a fila. Esta seção descreve como confirmar se uma fila está vazia.

Para confirmar se uma fila está vazia (console)

1. Interrompa o envio de mensagens por todos os produtores.
2. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
3. No painel de navegação, escolha Queues.
4. Na página Queues (Filas), escolha uma fila.
5. Escolha a guia Monitoring (Monitoramento).
6. No canto superior direito dos painéis de monitoramento, escolha a seta para baixo ao lado do símbolo Refresh (Atualizar). No menu suspenso, escolha Auto refresh (Atualização automática). Deixe Refresh interval (Atualização do intervalo) como 1 Minute (1 minuto).

7. Observe os seguintes painéis:

   - Número aproximado de mensagens atrasadas
   - Número aproximado de mensagens não visíveis
   - Número aproximado de mensagens visíveis

   Quando todos eles mostram valores `0` por vários minutos, a fila está vazia.

Para confirmar que uma fila está vazia (AWS CLI, AWS API)

1. Interrompa o envio de mensagens por todos os produtores.
2. Execute repetidamente um dos seguintes comandos:

   - AWS CLI: `get-queue-attributes`
   - AWS API: `GetQueueAttributes`
3. Observe as métricas dos seguintes atributos:

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

   Quando todos eles são `0` por vários minutos, a fila está vazia.

Se você confia nas CloudWatch métricas da Amazon, certifique-se de ver vários pontos de dados zero consecutivos antes de considerar a fila vazia. Para obter mais informações sobre CloudWatch métricas, consulte CloudWatch Métricas disponíveis para o Amazon SQS.

Excluir uma fila do Amazon SQS

Se você não usa mais uma fila do Amazon SQS e não prevê usá-la em um futuro próximo, recomendamos excluí-la.

> **Tip**
>
> Se você quiser verificar se uma fila está vazia antes de excluí-la, consulte Confirmando que uma fila do Amazon SQS está vazia.

Você pode excluir uma fila, mesmo quando ela não estiver vazia. Para excluir as mensagens em uma fila, mas não a própria fila, limpe a fila.

Para excluir uma fila (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Na página Queues (Filas), escolha a fila a ser excluída.
4. Escolha Excluir.
5. Na caixa de diálogo Delete queue (Excluir fila), confirme a exclusão inserindo **`delete`**.
6. Escolha Excluir.

Para excluir uma fila (AWS CLI e API)

Você pode usar um dos seguintes comandos para excluir uma fila:

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

Limpando mensagens de uma fila usando o console do Amazon SQS

Se não quiser excluir uma fila do Amazon SQS, mas precisar excluir todas as mensagens dela, limpe a fila. O processo de exclusão de mensagens pode levar até 60 segundos. Recomendamos aguardar 60 segundos, qualquer que seja o tamanho da fila.

> **Important**
>
> Quando você limpar uma fila, não poderá recuperar nenhuma mensagem excluída.

Para limpar uma fila (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Na página Queues (Filas), escolha a fila a ser limpa.
4. Em Actions (Ações), escolha Purge (Limpar).
5. Na caixa de diálogo Purge queue (Limpar fila), confirme a limpeza inserindo **`purge`** e escolhendo Purge (Limpar).

   Todas as mensagens são removidas da fila. O console exibe uma banner de confirmação.

Tarefas comuns para começar a usar o Amazon SQS

Agora que você criou uma fila e aprendeu como enviar, receber e excluir mensagens e como excluir uma fila, você pode querer experimentar o seguinte:

- Para acionar uma função do Lambda, consulte Configurando uma fila do Amazon SQS para acionar uma função AWS Lambda.
- Saiba como configurar filas, incluindo SSE e outros recursos.
- Saiba como enviar uma mensagem com atributos.
- Saiba como enviar uma mensagem de uma VPC.
- Para saber mais sobre a funcionalidade e a arquitetura do Amazon SQS, consulte Tipos de fila do Amazon SQS e Arquitetura básica do Amazon SQS.
- Para conhecer as diretrizes e advertências que ajudarão você a aproveitar ao máximo o Amazon SQS, consulte Práticas recomendadas do Amazon SQS.
- Explore os exemplos do Amazon SQS para um dos AWS SDKs, como o Developer Guide.AWS SDK for Java 2.x
- Para saber mais sobre os comandos do Amazon SQS, consulte a AWS CLI Referência de AWS CLI comandos.
- Para saber mais sobre as ações do Amazon SQS, consulte a Referência da API do Amazon Simple Queue Service.
- Aprenda a interagir com o Amazon SQS de forma programática: leia Como trabalhar com APIs e explore o Centro de Desenvolvimento:AWS
  - Java

- JavaScript
- PHP
- Python
- Ruby
- Windows e .NET

- Saiba mais sobre como acompanhar custos e recursos na seção Solução de problemas no Amazon SQS.
- Saiba mais sobre como proteger seus dados e o acesso aos dados na seção Segurança.
- Saiba mais sobre o fluxo de trabalho do Amazon SQS na seção Fluxo de trabalho do processo de controle de acesso do Amazon SQS.

Conceitos básicos de filas padrão do Amazon SQS

O Amazon SQS oferece padrão como o tipo de fila padrão. As filas padrão oferecem suporte a um número quase ilimitado de chamadas de API por segundo, por ação de API (`SendMessage`, `ReceiveMessage` ou `DeleteMessage`). As filas padrão oferecem suporte à entrega de at-least-once mensagens. No entanto, às vezes (por causa da arquitetura altamente distribuída que permite uma taxa de transferência praticamente ilimitada), mais de uma cópia da mensagem pode ser entregue fora de ordem. As filas padrão oferecem a melhor ordenação possível, o que garante a entrega das mensagens normalmente na mesma ordem em que foram enviadas.

O Amazon SQS armazena redundantemente uma mensagem em mais de uma zona de disponibilidade (AZ) antes de `SendMessage` ser reconhecido. Como as cópias das mensagens são armazenadas em várias AZs, nenhuma falha em um único computador, rede ou AZ pode tornar as mensagens inacessíveis.

Para obter informações sobre como criar e configurar filas usando o console do Amazon SQS, consulte Crie uma fila usando o console do Amazon SQS. Para exemplos de Java, consulte Exemplos de SDK do Java do Amazon SQS.

É possível usar filas de mensagens padrão em vários cenários, contanto que a aplicação possa processar as mensagens que chegam mais de uma vez e fora de ordem, por exemplo:

- Desacoplar solicitações do usuário em tempo real de trabalhos intensos em segundo plano: permitir que os usuários façam upload de mídia redimensionando-a ou codificando-a.
- Alocar tarefas para nós com vários operadores: processar um alto número de solicitações de validação de cartão de crédito.
- Mensagens em lotes para processamento futuro: programar várias entradas para adicioná-las ao banco de dados.

Para cotas relacionadas a filas padrão, consulte Cotas.

Para as práticas recomendadas ao trabalhar com filas padrão, consulte Recomendações para filas FIFO e padrão do Amazon SQS.

Ordenação de mensagens

Uma fila padrão faz o possível para preservar a ordem das mensagens, porém mais de uma cópia de uma mensagem pode ser entregue fora de ordem. Se o sistema exigir que a ordem seja preservada, recomendamos usar uma FIFO (primeiro a entrar, primeiro a sair) ou adicionar informações sobre o sequenciamento em cada mensagem para que você possa reordená-las quando elas forem recebidas.

Uma t-least-once entrega

O Amazon SQS armazena cópias de suas mensagens em vários servidores para obter redundância e alta disponibilidade. Em raras ocasiões, um dos servidores que armazena a cópia de uma mensagem poderá ficar indisponível quando você receber ou excluir uma mensagem.

Se isso ocorrer, a cópia da mensagem não será excluída no servidor que está indisponível e você poderá obter essa cópia da mensagem novamente ao receber mensagens. Projete aplicativos para serem idempotentes (para não serem afetados quando a mesma mensagem é processada mais de uma vez).

Identificadores de mensagens e filas do Amazon SQS

Esta seção descreve os identificadores de filas padrão e FIFO. Esses identificadores podem ajudar a localizar e manipular filas e mensagens específicas.

Identificadores de filas padrão do Amazon SQS

Para obter mais informações sobre os seguintes identificadores, consulte a Referência da API do Amazon Simple Storage Service.

Nome e URL da fila

Ao criar uma nova fila, você deve especificar o nome de uma fila exclusivo para sua conta e região da AWS . O Amazon SQS atribui a cada fila que você cria um identificador chamado URL da fila, que inclui o nome da fila e outros componentes do Amazon SQS. Sempre que você desejar executar uma ação em uma fila, forneça o URL da fila.

O seguinte URL é de uma fila chamada `MyQueue`, de propriedade de um usuário com o número de conta da AWS `123456789012`.

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

É possível recuperar o URL de uma fila programaticamente listando as filas e analisando a string que segue o número da conta. Para ter mais informações, consulte `ListQueues`.

ID de mensagem

Cada mensagem recebe um ID da mensagem atribuído pelo sistema que o Amazon SQS retorna para você na resposta `SendMessage`. Esse identificador é útil para identificar mensagens. O tamanho máximo de um ID de mensagem é 100 caracteres.

Identificador de recebimento

Toda vez que você recebe uma mensagem de uma fila, recebe um identificador de recebimento dessa mensagem. Esse identificador é associado à ação de recebimento da mensagem, e não à mensagem. Para excluir a mensagem ou alterar a visibilidade da mensagem, você deve fornecer o identificador de recebimento (não o ID de mensagem). Desse modo, você sempre deve receber uma mensagem antes de excluí-la (você não pode colocar uma mensagem na fila e, em seguida, recuperá-la). O tamanho máximo de um identificador de recebimento é 1024 caracteres.

> **Important**
>
> Se você receber uma mensagem mais de uma vez, cada vez que recebê-la, obterá um identificador de recebimento diferente. Você deve fornecer o identificador de recebimento recebido mais recentemente ao solicitar a exclusão da mensagem (caso contrário, a mensagem pode não ser excluída).

Veja a seguir um exemplo de um identificador de recebimento (quebrado em três linhas).

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

Cotas

A tabela a seguir lista as cotas relacionadas a filas padrão.

Quota	Descrição
Fila de atraso	O atraso padrão (mínimo) para uma fila é 0 segundo. O máximo é 15 minutos.
Filas listadas	1.000 filas por solicitação `ListQueues` .
Tempo de espera da sondagem longa	O tempo máximo de espera de sondagem longa é de 20 segundos.
Mensagens por fila (backlog)	O número de mensagens que uma fila do Amazon SQS pode armazenar é ilimitado.
Mensagens por fila (em andamento)	Para a maioria das filas padrão (dependendo do tráfego da fila e da lista de pendências), pode haver um máximo de aproximadamente 120.000 mensagens em trânsito (recebidas de uma fila por um consumidor, mas ainda não excluídas da fila). Se você atingir essa cota ao usar a sondagem curta, o Amazon SQS retornará a mensagem de erro `OverLimit` . Se você usar a sondagem longa, o Amazon SQS não retornará nenhuma mensagem de erro. Para evitar atingir o quota, você deve excluir mensagens da fila depois de serem processadas. Você também pode aumentar o número de filas que usar para processar as mensagens. Para solicitar um aumento, envie um pedido de suporte.
Nome da fila	O nome da fila pode ter até 80 caracteres. Os seguintes caracteres são aceitos: caracteres alfanuméricos, hifens (-) e sublinhados (_). **Note** Os nomes de fila diferenciam maiúsculas e minúsculas (por exemplo, `Test-queue` e `test-queue` são filas diferentes).

Quota	Descrição
Tag de fila	Não recomendamos adicionar mais de 50 tags a uma fila. A marcação é compatível com caracteres Unicode em UTF-8.
	A tag `Key` é necessária, mas a tag `Value` é opcional.
	A tag `Key` e a tag `Value` diferenciam maiúsculas de minúsculas.
	A tag `Key` e a tag `Value` podem incluir caracteres alfanuméricos Unicode em UTF-8 e espaços em branco. Os seguintes caracteres especiais são permitidos: `_ . : / = + - @`
	A tag `Key` ou `Value` não pode incluir o prefixo reservado `aws:` (não é possível excluir chaves ou valores de tag com esse prefixo).
	O comprimento máximo da tag `Key` é de 128 caracteres Unicode em UTF-8. A tag `Key` não pode estar vazia nem ser nula.
	O comprimento máximo da tag `Value` é de 256 caractere s Unicode em UTF-8. A tag `Value` pode estar vazia ou ser nula.
	As ações de marcação são limitadas a 30 TPS por. Conta da AWS Se a sua aplicação exigir uma taxa de transferê ncia mais alta, envie uma solicitação.

Introdução às filas FIFO no Amazon SQS

As filas FIFO (primeiro a entrar, primeiro a sair) têm todos os recursos das filas comuns, mas são projetadas para aprimorar o sistema de mensagens entre aplicações quando a ordem das operações e dos eventos é crucial ou quando duplicatas não podem ser toleradas.

Exemplos de situações em que você pode usar filas FIFO incluem os seguintes:

1. Sistema de gerenciamento de pedidos de comércio eletrônico em que o pedido é essencial
2. Integração com sistemas de terceiros em que os eventos precisam ser processados em ordem
3. Processamento de entradas inseridas pelo usuário no pedido inserido
4. Comunicações e redes: envio e recebimento de dados e informações na mesma ordem
5. Sistemas de computador: garantir que os comandos inseridos pelo usuário sejam executados na ordem correta.
6. Institutos educacionais: impedir que um aluno se matricule em um curso antes de criar uma conta.
7. Sistema de emissão de tíquetes online: no qual os tíquetes são distribuídos por ordem de chegada

> **Note**
>
> As filas FIFO também fornecem processamento exatamente uma vez, mas têm um número limitado de transações por segundo (TPS). É possível usar o modo de alto throughput do Amazon SQS com a fila FIFO para aumentar o limite de transações. Para obter detalhes sobre como usar o modo de alto throughput, consulte Alta taxa de transferência para filas FIFO no Amazon SQS. Para obter mais informações sobre cotas de taxa de transferência, consulte the section called “Cotas de mensagens”.

As filas FIFO do Amazon SQS estão disponíveis em todas as regiões em que o Amazon SQS está disponível

Para saber mais sobre como usar filas FIFO com pedidos complexos, consulte Solving Complex Ordering Challenges with Amazon SQS FIFO Queues (Resolver desafios complexos de pedidos com filas FIFO do Amazon SQS).

Para obter informações sobre como criar e configurar filas usando o console do Amazon SQS, consulte Crie uma fila usando o console do Amazon SQS. Para exemplos de Java, consulte Exemplos de SDK do Java do Amazon SQS.

Para as práticas recomendadas ao trabalhar com filas FIFO, consulte Recomendações adicionais para filas FIFO do Amazon SQS e Recomendações para filas FIFO e padrão do Amazon SQS.

Lógica de entrega de filas FIFO no Amazon SQS

Os conceitos a seguir podem ajudar a entender melhor o envio e o recebimento de mensagens por FIFO.

Enviar mensagens

Se várias mensagens forem enviadas em sucessão a uma fila FIFO, cada uma com um ID de eliminação de duplicação de mensagem distinto, o Amazon SQS armazenará as mensagens e confirmará a transmissão. Em seguida, cada mensagem pode ser recebida e processada na ordem exata em que as mensagens foram transmitidas.

Em filas FIFO, as mensagens são ordenadas com base no ID de grupo de mensagens. Se vários hosts (ou threads diferentes no mesmo host) enviarem mensagens com o mesmo ID de grupo de mensagens para uma fila FIFO, o Amazon SQS armazenará as mensagens na ordem de chegada para processamento. Para garantir que o Amazon SQS preserve a ordem na qual as mensagens são enviadas e recebidas, cada produtor deve usar um ID de grupo de mensagens exclusivo para enviar todas as suas próprias mensagens.

A lógica da fila FIFO aplica-se apenas para cada ID de grupo de mensagens. Cada ID de grupo de mensagens representa um grupo de mensagens ordenadas distinto em uma fila do Amazon SQS. Para cada ID de grupo de mensagens, todas as mensagens são enviadas e recebidas na ordem estrita. No entanto, as mensagens com valores de ID de grupo de mensagens diferentes podem ser enviadas e recebidas fora de ordem. Você deve associar um ID de grupo de mensagens a uma mensagem. Se você não fornecer um ID de grupo de mensagens, a ação resultará em falha. Se você precisar de um único grupo de mensagens ordenadas, forneça o mesmo ID de grupo de mensagens para as mensagens enviadas para a fila FIFO.

Recebimento de mensagens

Você não pode solicitar o recebimento de mensagens com um ID de grupo de mensagens específico.

Ao receber mensagens de uma fila FIFO com vários IDs de grupo de mensagens, o Amazon SQS primeiro tenta retornar o máximo possível de mensagens com o mesmo ID de grupo de mensagens. Isso permite que outros clientes processem mensagens com um ID de grupo de mensagens diferente. Quando você recebe uma mensagem com um ID de grupo de mensagens, nenhuma outra mensagem para o mesmo ID de grupo de mensagens são retornadas, a menos que você exclua a mensagem ou ela se torne visível.

> **Note**
>
> É possível receber até 10 mensagens em uma única chamada usando o parâmetro de solicitação `MaxNumberOfMessages` da `ReceiveMessage` ação de Essas mensagens retêm a ordem de FIFO e podem ter o mesmo ID de grupo de mensagens. Assim, se houver menos de 10 mensagens disponíveis com o mesmo ID de grupo de mensagens, você poderá receber mensagens de outro ID de grupo de mensagens, no mesmo lote de 10 mensagens, mas ainda na ordem de FIFO.

Repetir várias vezes

As filas FIFO permitem várias tentativas ao produtor ou ao consumidor:

- Se o produtor detectar uma falha de ação `SendMessage`, ele poderá tentar enviá-la novamente quantas vezes for necessário, usando o mesmo ID de eliminação de duplicação de mensagens. Supondo que o produtor receba pelo menos uma confirmação antes que o intervalo de eliminação de duplicação expire, as várias tentativas não afetarão a ordenação das mensagens nem introduzirão duplicatas.
- Se o consumidor detectar uma falha de ação `ReceiveMessage`, ele poderá tentar enviá-la novamente quantas vezes for necessário, usando o mesmo ID de tentativa de solicitação de recebimento. Supondo que o consumidor receba pelo menos uma confirmação antes do tempo limite de visibilidade expirar, as várias tentativas não afetarão a ordenação das mensagens.
- Quando você recebe uma mensagem com um ID de grupo de mensagens, nenhuma outra mensagem para o mesmo ID de grupo de mensagens são retornadas, a menos que você exclua a mensagem ou ela se torne visível.

Ordenação de mensagens de fila FIFO no Amazon SQS

A fila FIFO aprimora e complementa a fila padrão. Os recursos mais importantes desse tipo de fila são entrega FIFO (primeiro a entrar, primeiro a sair) e processamento exatamente uma vez:

- A ordem na qual as mensagens são enviadas e recebidas é estritamente preservada e uma mensagem é entregue uma vez e permanece indisponível até que um consumidor a processe e exclua.
- As duplicações não são introduzidas na fila.

As filas FIFO também oferecem suporte a grupos de mensagens, que permitem diversos grupos de mensagens ordenadas em uma única fila. Não há cota para o número de grupos de mensagens dentro de uma fila FIFO.

Processamento de exatamente uma vez no Amazon SQS

Ao contrário das filas padrão, as filas FIFO não introduzem mensagens duplicadas. As filas FIFO ajudam a evitar o envio de duplicações para uma fila. Se você tentar novamente a ação `SendMessage` dentro do intervalo de eliminação de duplicação de 5 minutos, o Amazon SQS não introduzirá duplicações na fila.

Para configurar a eliminação de duplicação, você deve realizar umas das seguintes ações:

- Ativar a eliminação de duplicação baseada em conteúdo. Isso instrui o Amazon SQS a usar um hash SHA-256 para gerar o ID de eliminação de duplicação de mensagens usando o corpo da mensagem, mas não os atributos dela. Para obter mais informações, consulte a documentação sobre ações `CreateQueue`, `GetQueueAttributes` e `SetQueueAttributes` na Referência da API do Amazon Simple Queue Service.
- Forneça explicitamente o ID de eliminação de duplicação da mensagem (ou visualize o número de sequência) para a mensagem. Para obter mais informações, consulte a documentação sobre ações `SendMessage`, `SendMessageBatch` e `ReceiveMessage` na Referência da API do Amazon Simple Queue Service.

Migração de uma fila padrão para uma fila FIFO no Amazon SQS

Se houver uma aplicação que usa filas padrão e você quiser aproveitar os recursos de ordenação ou de processamento exatamente uma vez das filas FIFO, precisará configurar a fila e sua aplicação corretamente.

> **Note**
>
> Não é possível converter uma fila padrão existente em uma fila FIFO. Para migrar, é necessário criar uma nova fila FIFO para a aplicação ou excluir a fila padrão atual e recriá-la como uma fila FIFO.

Use a seguinte lista de verificação para garantir que sua aplicação funcione corretamente com uma fila FIFO:

- Use o modo de alto throughput recomendado para FIFO a fim de obter maior throughput. Para saber mais sobre cotas de mensagens, consulte Cotas de mensagens do Amazon SQS.
- As filas FIFO não dão suporte a atrasos por mensagem, apenas a atrasos por fila. Se seu aplicativo define o mesmo valor do parâmetro `DelaySeconds` em cada mensagem, você deve modificar o aplicativo para remover o atraso por mensagem e definir `DelaySeconds` em toda a fila.
- O grupo de mensagens é um atributo FIFO exclusivo que permite que os clientes processem mensagens em paralelo enquanto mantêm os respectivos pedidos. Os clientes organizam as mensagens em grupos especificando um ID do grupo de mensagens. Os grupos de mensagens geralmente se baseiam em uma dimensão comercial para determinada workload. Para escalar melhor com as filas FIFO, use uma dimensão comercial mais detalhada para o ID da mensagem. Quanto maior o número de IDs de grupos de mensagens para os quais você distribui mensagens, maior o número de mensagens que a FIFO disponibiliza para consumo.
- Antes de enviar mensagens para uma fila FIFO, confirme o seguinte:
  - Se seu aplicativo pode enviar mensagens com corpos de mensagem idênticos, você pode modificar o aplicativo para fornecer um ID de eliminação de duplicação de mensagem exclusivo para cada mensagem enviada.
  - Se seu aplicativo envia mensagens com corpos de mensagem exclusivos, você pode ativar a eliminação de duplicação baseada em conteúdo.
- Você não precisa fazer alterações de código para seu consumidor. No entanto, se levar muito tempo para processar mensagens e o tempo limite de visibilidade for definido como um valor alto, considere a adição de um ID de tentativa de solicitação de recebimento a cada ação `ReceiveMessage`. Isso permite que você repita tentativas de recebimento em caso de falhas de rede e impede que as filas pausem devido a tentativas de recebimento com falha.

Para obter mais informações, consulte a Referência da API do Amazon Simple Queue Service.

Alta taxa de transferência para filas FIFO no Amazon SQS

As filas FIFO de alta taxa de transferência no Amazon SQS gerenciam com eficiência a alta taxa de transferência de mensagens enquanto mantêm uma ordem rígida de mensagens, garantindo confiabilidade e escalabilidade para aplicativos que processam várias mensagens. Essa solução é ideal para cenários que exigem alto rendimento e entrega ordenada de mensagens.

As filas FIFO de alta taxa de transferência do Amazon SQS não são necessárias em cenários em que a ordenação estrita de mensagens não é crucial e em que o volume de mensagens recebidas é relativamente baixo ou esporádico. Por exemplo, se você tiver um aplicativo de pequena escala que processa mensagens pouco frequentes ou não sequenciais, a complexidade e o custo adicionais associados às filas FIFO de alto rendimento podem não ser justificados. Além disso, se seu aplicativo não exigir os recursos aprimorados de taxa de transferência fornecidos pelas filas FIFO de alta taxa de transferência, optar por uma fila padrão do Amazon SQS pode ser mais econômico e mais simples de gerenciar.

Para aumentar a capacidade de solicitação em filas FIFO de alto rendimento, é recomendável aumentar o número de grupos de mensagens. Para ter mais informações sobre cotas de mensagens de throughput alto, consulte Amazon SQS service quotas no Referência geral da Amazon Web Services.

Para obter informações sobre cotas por fila e estratégias de distribuição de dados, consulte e. Cotas de mensagens do Amazon SQS Partições e distribuição de dados para alta taxa de transferência para filas FIFO do SQS

Tópicos

- Casos de uso de alta taxa de transferência para filas FIFO do Amazon SQS
- Partições e distribuição de dados para alta taxa de transferência para filas FIFO do SQS
- Habilite alta taxa de transferência para filas FIFO no Amazon SQS

Casos de uso de alta taxa de transferência para filas FIFO do Amazon SQS

Os casos de uso a seguir destacam as diversas aplicações de filas FIFO de alto rendimento, mostrando sua eficácia em todos os setores e cenários:

1. Processamento de dados em tempo real: aplicativos que lidam com fluxos de dados em tempo real, como processamento de eventos ou ingestão de dados de telemetria, podem se beneficiar de filas FIFO de alto rendimento para lidar com o fluxo contínuo de mensagens e, ao mesmo tempo, preservar sua ordem para uma análise precisa.
2. Processamento de pedidos de comércio eletrônico: em plataformas de comércio eletrônico em que manter a ordem das transações do cliente é fundamental, as filas FIFO de alto rendimento garantem que os pedidos sejam processados sequencialmente e sem atrasos, mesmo durante os períodos de pico de compras.
3. Serviços financeiros: as instituições financeiras que lidam com dados comerciais ou transacionais de alta frequência dependem de filas FIFO de alto rendimento para processar dados e transações de mercado com latência mínima, ao mesmo tempo em que cumprem os rígidos requisitos regulatórios para pedidos de mensagens.
4. Streaming de mídia: plataformas de streaming e serviços de distribuição de mídia utilizam filas FIFO de alto rendimento para gerenciar a entrega de arquivos de mídia e conteúdo de streaming, garantindo experiências de reprodução suaves para os usuários e mantendo a ordem correta de entrega do conteúdo.

Partições e distribuição de dados para alta taxa de transferência para filas FIFO do SQS

O Amazon SQS armazena dados da fila FIFO em partições. Uma partição é uma alocação de armazenamento para uma fila que é automaticamente replicada em várias zonas de disponibilidade em uma região. AWS Você não gerencia partições. Em vez disso, o Amazon SQS lida com o gerenciamento de partições

Para filas FIFO, o Amazon SQS modifica o número de partições em uma fila nas seguintes situações:

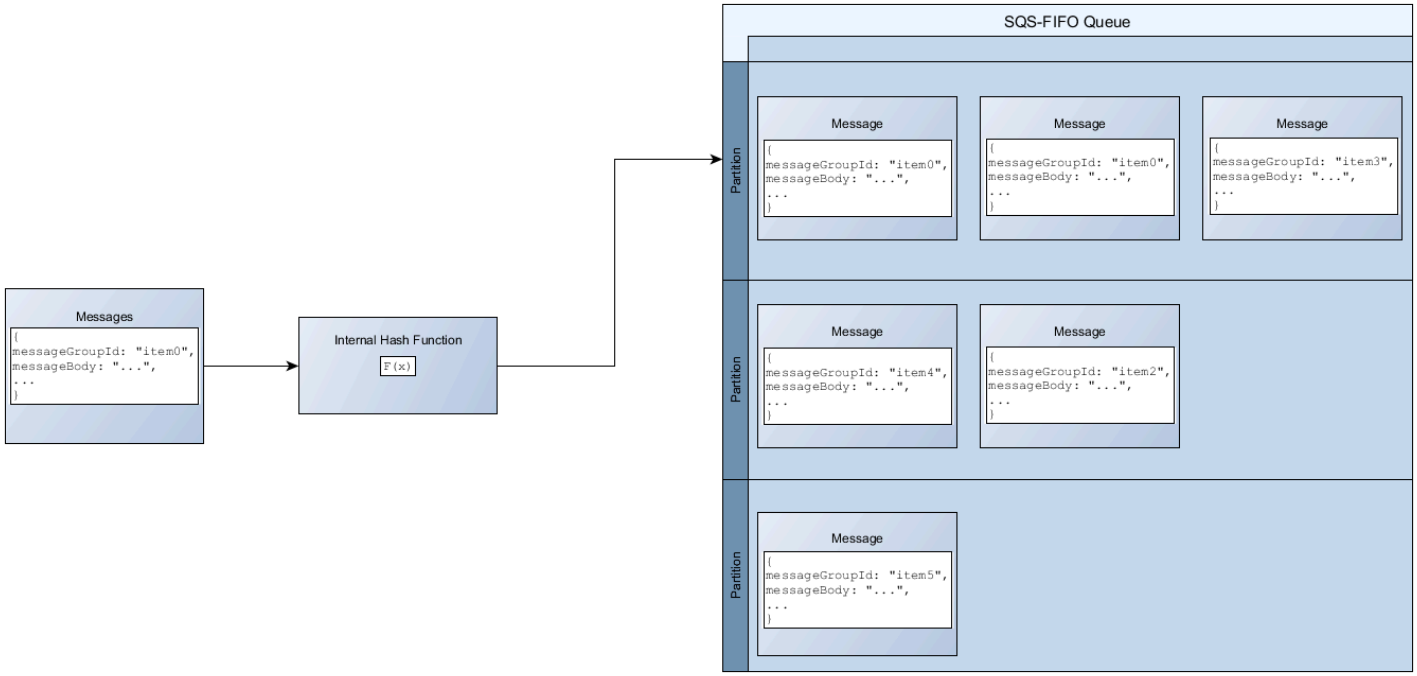
- Se a taxa de solicitação atual se aproximar ou exceder o que as partições existentes podem suportar, partições adicionais serão alocadas até que a fila atinja a cota regional. Para obter informações sobre cotas, consulte Cotas de mensagens do Amazon SQS.
- Se as partições atuais tiverem baixa utilização, o número de partições poderá ser reduzido.

O gerenciamento de partições ocorre automaticamente em segundo plano e é transparente para as aplicações. Sua fila e mensagens estão disponíveis em todos os momentos.

Distribuindo dados por IDs de grupo de mensagens

Para adicionar uma mensagem a uma fila FIFO, o Amazon SQS usa o valor do ID do grupo de mensagens de cada mensagem como entrada para uma função de hash interna. O valor de saída da função de hash determina a partição na qual a mensagem será armazenada.

O diagrama a seguir mostra uma fila que abrange várias partições. O ID do grupo de mensagens da fila é baseado no número do item. O Amazon SQS usa sua função de hash para determinar onde armazenar um novo item, neste caso, com base no valor de hash da string `item0`. Observe que os itens são armazenados na mesma ordem em que são adicionados à fila. A localização de cada item é determinada pelo valor de hash de seu ID de grupo de mensagens.



> **Note**
>
> O Amazon SQS é otimizado para distribuição uniforme de itens nas partições de uma fila FIFO, independentemente do número de partições. AWS recomenda que você use IDs de grupos de mensagens que possam ter um grande número de valores distintos.

Otimizando a utilização de partições

Cada partição suporta até 3.000 mensagens por segundo com processamento em lote ou até 300 mensagens por segundo para operações de envio, recebimento e exclusão em regiões compatíveis.

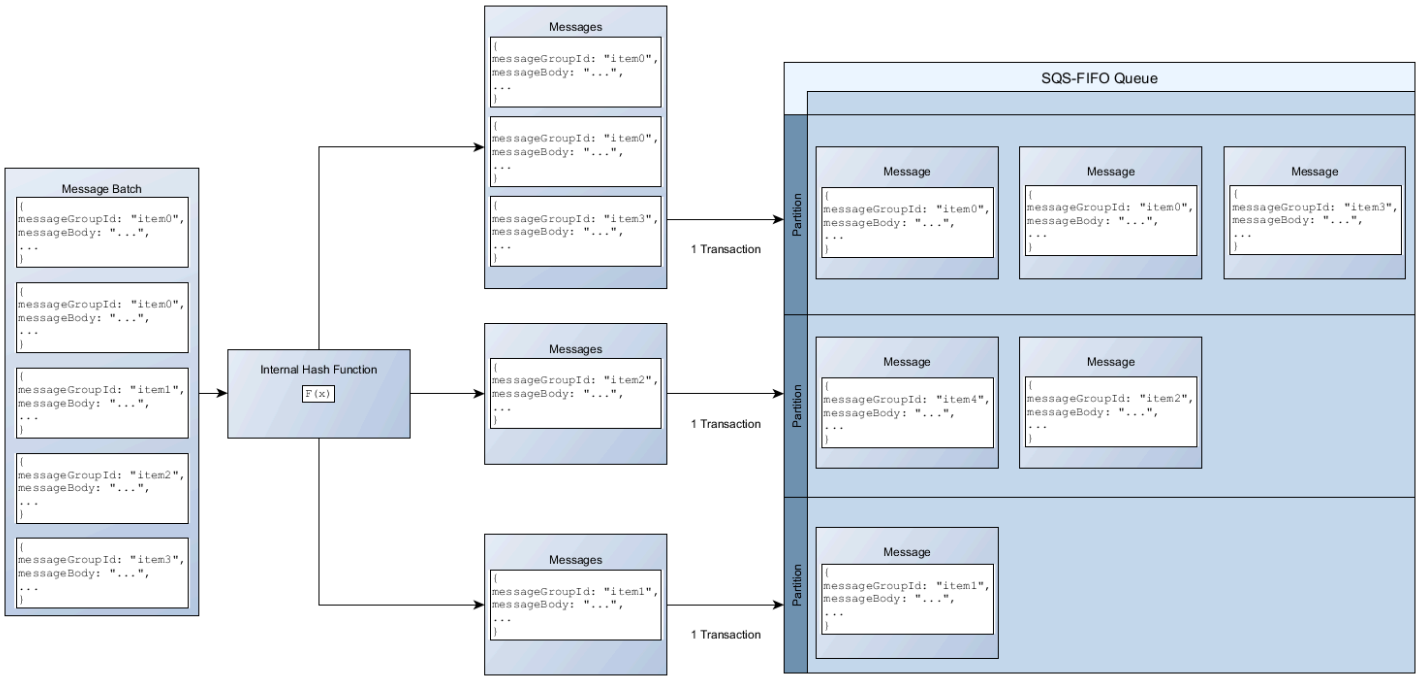
Para ter mais informações sobre cotas de mensagens de throughput alto, consulte Amazon SQS service quotas no Referência geral da Amazon Web Services.

Ao usar APIs em lote, cada mensagem é encaminhada com base no processo descrito em Distribuindo dados por IDs de grupo de mensagens. Mensagens encaminhadas para a mesma partição são agrupadas e processadas em uma única transação.

Para otimizar a utilização da partição para a `SendMessageBatch` API, AWS recomenda agrupar mensagens em lote com os mesmos IDs de grupo de mensagens sempre que possível.

Para otimizar a utilização da partição para as `ChangeMessageVisibilityBatch` APIs `DeleteMessageBatch` e, AWS recomenda usar `ReceiveMessage` solicitações com o `MaxNumberOfMessages` parâmetro definido como 10 e agrupar em lotes os identificadores de recebimento retornados por uma única solicitação. `ReceiveMessage`

No exemplo a seguir, um lote de mensagens com vários IDs de grupo de mensagens é enviado. O lote é dividido em três grupos, com cada um contando para a cota da partição.



> **Note**
>
> O Amazon SQS garante somente que mensagens com a mesma função de hash interna do ID de grupo de mensagens sejam agrupadas em uma solicitação em lote. Dependendo da saída da função hash interna e do número de partições, mensagens com diferentes IDs de

grupo de mensagens podem ser agrupadas. Como a função hash ou o número de partições pode ser alterado a qualquer momento, as mensagens agrupadas em um ponto podem não ser agrupadas posteriormente.

Habilite alta taxa de transferência para filas FIFO no Amazon SQS

Você pode habilitar a alta taxa de transferência para qualquer fila FIFO nova ou existente. O recurso inclui três novas opções ao criar e editar filas FIFO:

- Enable high throughput FIFO (Habilitar FIFO de alta taxa de transferência): aumenta a taxa de transferência disponível para mensagens na fila FIFO atual.
- Deduplication scope (Escopo de eliminação de duplicação): especifica se a eliminação de duplicação ocorre no nível da fila ou do grupo de mensagens.
- FIFO throughput limit (Limite de taxa de transferência FIFO): especifica se a cota de taxa de transferência em mensagens na fila FIFO está definida no nível da fila ou do grupo de mensagens.

Para habilitar a alta taxa de transferência para uma fila FIFO (console)

1. Inicie criando ou editando uma fila FIFO.
2. Ao especificar opções para a fila, escolha Enable high throughput FIFO (Habilitar FIFO de alta taxa de transferência).

   Habilitar a alta taxa de transferência para filas FIFO define as opções relacionadas da seguinte maneira:

   - Deduplication scope (Escopo de eliminação de duplicação) é definido como Message group (Grupo de mensagens), a configuração necessária para usar a alta taxa de transferência para filas FIFO.
   - FIFO throughput limit (Limite de taxa de transferência FIFO) é definido como Per message group ID (Por ID do grupo de mensagens), a configuração necessária para usar a alta taxa de transferência para filas FIFO.

   Se você alterar qualquer uma das configurações necessárias para usar a alta taxa de transferência para filas FIFO, a taxa de transferência normal estará em vigor para a fila e a eliminação de duplicação ocorrerá conforme especificado.

3. Continue especificando todas as opções para a fila. Ao concluir, escolha Create queue (Criar fila) ou Save (Salvar).

Depois de criar ou editar a fila FIFO, você pode enviar mensagens para ela e receber e excluir mensagens, tudo em um TPS mais alto. Para cotas de alto throughput, consulte Throughput de mensagens em Cotas de mensagens do Amazon SQS.

Termos-chave do Amazon SQS

Os seguintes termos-chave podem ajudar você a entender melhor a funcionalidade das filas FIFO. Para obter mais informações, consulte a Referência da API do Amazon Simple Queue Service.

ID de eliminação de duplicação de mensagens

O token usado para a eliminação de duplicação de mensagens enviadas. Se uma mensagem com um ID de eliminação de duplicação de mensagens específico for enviada com êxito, todas as mensagens enviadas com o mesmo ID de eliminação de duplicação de mensagens serão aceitas com êxito, mas não serão entregues durante o intervalo de eliminação de duplicação de cinco minutos.

> **Note**
>
> O Amazon SQS continua acompanhando o ID de eliminação de duplicação da mensagem mesmo depois que a mensagem é recebida e excluída.

ID do grupo de mensagens

A marcação que especifica que uma mensagem pertence a um grupo de mensagens específico. As mensagens que pertencem ao mesmo grupo de mensagens são sempre processadas uma a uma, em uma ordem estrita relativa ao grupo de mensagens (no entanto, as mensagens que pertencem a diferentes grupos de mensagens podem ser processadas fora de ordem).

ID de tentativa de solicitação de recebimento

O token usado para a eliminação da duplicação de chamadas de `ReceiveMessage`.

Número de sequência

O número grande e não consecutivo que o Amazon SQS atribui a cada mensagem.

Compatibilidade com FIFO no Amazon SQS

Clientes

Atualmente, o cliente assíncrono no buffer do Amazon SQS não oferece suporte a filas FIFO.

Serviços

Se seu aplicativo usa vários AWS serviços ou uma combinação de AWS serviços externos, é importante entender qual funcionalidade de serviço não é compatível com filas FIFO.

Alguns serviços AWS ou serviços externos que enviam notificações para o Amazon SQS podem não ser compatíveis com filas FIFO, apesar de permitirem que você defina uma fila FIFO como destino.

Atualmente, os seguintes recursos dos AWS serviços não são compatíveis com filas FIFO:

- Notificações de eventos do Amazon S3
- Ganchos do ciclo de vida do Auto Scaling
- AWS IoT Ações de regras
- AWS Lambda Dead Letter Queues

Para obter informações sobre a compatibilidade de outros produtos com filas FIFO, consulte a documentação do seu produto.

Identificadores de fila e mensagem FIFO no Amazon SQS

Esta seção descreve os identificadores de filas FIFO. Esses identificadores podem ajudar a localizar e manipular filas e mensagens específicas.

Tópicos

Identificadores para filas FIFO no Amazon SQS

Para obter mais informações sobre os seguintes identificadores, consulte a Referência da API do Amazon Simple Storage Service.

Nome e URL da fila

Ao criar uma nova fila, você deve especificar o nome de uma fila exclusivo para sua conta e região da AWS . O Amazon SQS atribui a cada fila que você cria um identificador chamado URL da fila, que inclui o nome da fila e outros componentes do Amazon SQS. Sempre que você desejar executar uma ação em uma fila, forneça o URL da fila.

O nome de uma fila FIFO deve terminar com o sufixo `.fifo`. O sufixo conta para a cota de 80 caracteres do nome da fila. Para determinar se uma fila é FIFO, você pode conferir se o nome da fila termina com o sufixo.

A seguir está o URL da fila para uma fila FIFO chamada de `MyQueue` propriedade de um usuário com o número da conta da AWS. `123456789012`

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

É possível recuperar o URL de uma fila programaticamente listando as filas e analisando a string que segue o número da conta. Para ter mais informações, consulte `ListQueues`.

ID de mensagem

Cada mensagem recebe um ID da mensagem atribuído pelo sistema que o Amazon SQS retorna para você na resposta `SendMessage`. Esse identificador é útil para identificar mensagens. O tamanho máximo de um ID de mensagem é 100 caracteres.

Identificador de recebimento

Toda vez que você recebe uma mensagem de uma fila, recebe um identificador de recebimento dessa mensagem. Esse identificador é associado à ação de recebimento da mensagem, e não à mensagem. Para excluir a mensagem ou alterar a visibilidade da mensagem, você deve fornecer o identificador de recebimento (não o ID de mensagem). Desse modo, você sempre deve receber uma mensagem antes de excluí-la (você não pode colocar uma mensagem na fila e, em seguida, recuperá-la). O tamanho máximo de um identificador de recebimento é 1024 caracteres.

> **Important**
>
> Se você receber uma mensagem mais de uma vez, cada vez que recebê-la, obterá um identificador de recebimento diferente. Você deve fornecer o identificador de recebimento

recebido mais recentemente ao solicitar a exclusão da mensagem (caso contrário, a mensagem pode não ser excluída).

Veja a seguir um exemplo de um identificador de recebimento (quebrado em três linhas).

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

Identificadores adicionais para filas FIFO do Amazon SQS

Para obter mais informações sobre os identificadores a seguir, consulte Processamento de exatamente uma vez no Amazon SQS e a Referência da API do Amazon Simple Queue Service.

ID de eliminação de duplicação de mensagens

O token usado para a eliminação de duplicação de mensagens enviadas. Se uma mensagem com um ID de eliminação de duplicação de mensagens específico for enviada com êxito, todas as mensagens enviadas com o mesmo ID de eliminação de duplicação de mensagens serão aceitas com êxito, mas não serão entregues durante o intervalo de eliminação de duplicação de cinco minutos.

ID do grupo de mensagens

A marcação que especifica que uma mensagem pertence a um grupo de mensagens específico. As mensagens que pertencem ao mesmo grupo de mensagens são sempre processadas uma a uma, em uma ordem estrita relativa ao grupo de mensagens (no entanto, as mensagens que pertencem a diferentes grupos de mensagens podem ser processadas fora de ordem).

Número de sequência

O número grande e não consecutivo que o Amazon SQS atribui a cada mensagem.

Cotas do Amazon SQS

Este tópico lista as cotas do Amazon Simple Queue Service (Amazon SQS).

Tópicos

Cotas de fila FIFO do Amazon SQS

Cotas do Amazon SQS

A tabela a seguir lista as cotas relacionadas a filas FIFO.

Cota	Descrição
Fila de atraso	O atraso padrão (mínimo) para uma fila é 0 segundo. O máximo é 15 minutos.
Filas listadas	1.000 filas por solicitação `ListQueues` .
Tempo de espera da sondagem longa	O tempo máximo de espera de sondagem longa é de 20 segundos.
Grupos de mensagens	Não há cota para o número de grupos de mensagens dentro de uma fila FIFO.
Mensagens por fila (backlog)	O número de mensagens que uma fila do Amazon SQS pode armazenar é ilimitado.
Mensagens por fila (em andamento)	Para filas FIFO, pode haver no máximo de 20.000 mensagens em trânsito (recebidas de uma fila por um consumidor, mas ainda não excluídas da fila). Se você atingir essa cota, o Amazon SQS não retornará nenhuma mensagem de erro.

Cota	Descrição
Nome da fila	O nome de uma fila FIFO deve terminar com o sufixo `.fifo`. O sufixo conta para a cota de 80 caracteres do nome da fila. Para determinar se uma fila é FIFO, você pode conferir se o nome da fila termina com o sufixo.
Tag de fila	Não recomendamos adicionar mais de 50 tags a uma fila. A marcação é compatível com caracteres Unicode em UTF-8.
	A tag `Key` é necessária, mas a tag `Value` é opcional.
	A tag `Key` e a tag `Value` diferenciam maiúsculas de minúsculas.
	A tag `Key` e a tag `Value` podem incluir caracteres alfanuméricos Unicode em UTF-8 e espaços em branco. Os seguintes caracteres especiais são permitidos: `_ . : / = + - @`
	A tag `Key` ou `Value` não pode incluir o prefixo reservado `aws:` (não é possível excluir chaves ou valores de tag com esse prefixo).
	O comprimento máximo da tag `Key` é de 128 caracteres Unicode em UTF-8. A tag `Key` não pode estar vazia nem ser nula.
	O comprimento máximo da tag `Value` é de 256 caractere s Unicode em UTF-8. A tag `Value` pode estar vazia ou ser nula.
	As ações de marcação são limitadas a 30 TPS por. Conta da AWS Se a sua aplicação exigir uma taxa de transferê ncia mais alta, envie uma solicitação.

Cotas de mensagens do Amazon SQS

A tabela a seguir lista as cotas relacionadas a mensagens.

Cota	Descrição
ID de mensagem em lote	O ID da mensagem em lote pode ter até 80 caractere s. Os seguintes caracteres são aceitos: caracteres alfanuméricos, hifens (-) e sublinhados (_).
Atributos de mensagens	Uma mensagem pode conter até 10 atributos de metadados.
Lote de mensagens	Uma única solicitação em lote de mensagens pode incluir um máximo de 10 mensagens. Para obter mais informações, consulte Configurando o cliente BufferedA sync AmazonSQS na seção Ações em lote do Amazon SQS.
Conteúdo da mensagem	Uma mensagem pode incluir apenas XML, JSON e texto não formatado. Os seguintes caracteres Unicode são permitidos: `#x9` | `#xA` | `#xD` | `#x20` até `#xD7FF` | `#xE000` até `#xFFFD` | `#x10000` até `#x10FFFF` Os caracteres não incluídos nesta lista serão rejeitados. Para obter mais informações, consulte a Especificação W3C para caracteres.
ID do grupo de mensagens	Consuma mensagens do backlog para evitar o acúmulo de um grande backlog de mensagens com o mesmo ID de grupo de mensagens. `MessageGroupId` é necessário para filas FIFO. Você não pode usá-lo para filas padrão. Você deve associar um `MessageGroupId` não vazio com uma mensagem. Se você não fornecer um `MessageGroupId` , a ação resultará em falha.

Cota	Descrição
	O tamanho máximo de `MessageGroupId` é 128 caracteres. Valores válidos: caracteres alfanuméricos e pontuação `(!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~)` .
Retenção da mensagem	Por padrão, uma mensagem será retida por 4 dias. A duração mínima é de 60 segundos (1 minuto). A configuração máxima é de 1.209.600 seconds (14 dias).
Taxa de transferência da mensagem	As filas padrão oferecem suporte a um número quase ilimitado de chamadas de API por segundo, por ação de API (`SendMessage` , `ReceiveMessage` ou `DeleteMessage`).
	Filas FIFO • As filas FIFO comportam uma cota de 300 transações por segundo, por ação de API (`SendMessage` , `ReceiveMessage` , e `DeleteMessage`). • Se você usa o agrupamento em lote, as filas FIFO comportam até 3 mil mensagens por segundo, por ação de API (`SendMessage` , `ReceiveMessage` e `DeleteMessage`). As 3 mil mensagens por segundo representam 300 chamadas de API, cada uma com um lote de 10 mensagens.

Cota	Descrição
	Alta taxa de transferência para filas FIFO • Sem o processamento em lote (`SendMessage`, `ReceiveMessage` e `DeleteMessage`), o alto throughput de filas FIFO processa até 70 mil transações por segundo, por ação de API, nas regiões Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon) e Europa (Irlanda). • Para as regiões Leste dos EUA (Ohio) e Europa (Frankfurt), o throughput padrão é de 18 mil transações por segundo por ação de API. • Para as regiões Ásia-Pacífico (Mumbai), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney) e Ásia-Pacífico (Tóquio), o throughput padrão é de 9 mil transações por segundo por ação de API. • Para Europa (Londres) e América do Sul (São Paulo), o throughput padrão é de 4.500 transações por segundo por ação de API. • Para a taxa de transferência máxima, aumente o número de IDs de grupo de mensagens que você usa para mensagens enviadas sem processamento em lotes. • Você pode aumentar o throughput para até 700 mil mensagens por segundo usando APIs de processamento em lote (`SendMessageBatch` e `DeleteMessageBatch`) nas regiões Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon) e Europa (Irlanda). As 700 mil mensagens por segundo representam 70 mil transações por segundo, cada uma com um lote de 10 mensagens. Para as regiões Europa (Frankfurt) e Leste dos EUA (Ohio), você pode receber até 180 mil mensagens por segundo usando APIs de processamento em lote. As

Cota	Descrição
	180 mil mensagens por segundo representam 18 mil transações por segundo, cada uma com um lote de 10 mensagens. Para as regiões Ásia-Pacífico (Mumbai), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney) e Ásia-Pacífico (Tóquio), você pode receber até 90 mil mensagens por segundo com o processamento em lote. Para obter a taxa de transferência máxima ao usar `SendMessageBatch` e `DeleteMessageBatch`, todas as mensagens em uma solicitação em lote devem usar o mesmo ID de grupo de mensagens. • Para as regiões Europa (Londres) e América do Sul (São Paulo), você pode receber até 45 mil mensagens por segundo com o processamento em lote. Para obter a taxa de transferência máxima ao usar `SendMessageBatch` e `DeleteMessageBatch`, todas as mensagens em uma solicitação em lote devem usar o mesmo ID de grupo de mensagens. • Em todas as outras AWS regiões, a taxa de transferência máxima é de 2.400 (sem lotes) ou 24.000 (usando lotes) mensagens por segundo, por ação da API. • Para solicitar um aumento de cota acima do limite da região, envie uma solicitação de suporte. • Para ter mais informações, consulte Partições e distribuição de dados para alta taxa de transferência para filas FIFO do SQS.
Temporizador de mensagem	O atraso padrão (mínimo) para uma mensagem é 0 segundo. O máximo é 15 minutos.

Cota	Descrição
Tamanho da mensagem	O tamanho mínimo da mensagem é de 1 byte (1 caractere). O comprimento máximo é 262.144 bytes (256 KiB). Para enviar mensagens maiores que 256 KiB, você pode usar a Amazon SQS Extended Client Library para Java e a Amazon SQS Extended Client Library for Python. Essa biblioteca permite que você envie uma mensagem do Amazon SQS que contenha uma referência à carga útil de uma mensagem no Amazon S3. O tamanho máximo de carga é 2 GB. **Note** Essa biblioteca estendida funciona somente para clientes síncronos.
Tempo limite de visibilidade da mensagem	O tempo limite de visibilidade padrão para uma mensagem é de 30 segundos. O mínimo é 0 segundo. O máximo é 12 horas.
Informações de política	A cota máxima é 8.192 bytes, 20 declarações, 50 principais ou 10 condições. Para ter mais informações, consulte Cotas da política do Amazon SQS.

Cotas da política do Amazon SQS

A tabela a seguir lista as cotas relacionadas a políticas.

Nome	Máximo
Bytes	8,192
Condições	10

Nome	Máximo
Principais	50
Declarações	20
Ações por declaração	7

Recursos e funcionalidades do Amazon SQS

O Amazon SNS fornece os recursos e as funcionalidades a seguir.

Tópicos

Usando filas de mensagens mortas no Amazon SQS

O Amazon SQS oferece suporte a filas de cartas mortas (DLQs), que as filas de origem podem direcionar para mensagens que não foram processadas com sucesso. As DLQs são úteis para depurar seu aplicativo porque você pode isolar mensagens não consumidas para determinar por que o processamento não foi bem-sucedido. Para um desempenho ideal, é uma prática recomendada manter a fila de origem e o DLQ dentro da mesma região Conta da AWS . Quando as mensagens estiverem em uma fila de mensagens mortas, você poderá:

- Examinar logs para encontrar as exceções que podem ter causado a entrega de mensagens a uma fila de mensagens mortas.
- Analise o conteúdo das mensagens movidas para a fila de mensagens mortas para diagnosticar problemas no aplicativo.

- Determinar se você concedeu ao consumidor tempo suficiente para processar mensagens.
- Mova as mensagens para fora da fila de mensagens mortas usando o redirecionamento da fila de mensagens mortas.

Você deve primeiro criar uma nova fila antes de configurá-la como uma fila de mensagens mortas. Para obter informações sobre como configurar uma fila de mensagens não entregues usando o console do Amazon SQS, consulte Saiba como configurar uma fila de mensagens sem saída usando o console do Amazon SQS. Para obter ajuda com filas de mensagens mortas, por exemplo, como configurar um alarme para qualquer mensagem movida para uma fila de mensagens mortas, consulte. Crie alarmes para filas de mensagens sem saída usando a Amazon CloudWatch

Usando políticas para filas de cartas mortas

Use uma política de redirecionamento para especificar o. `maxReceiveCount` `maxReceiveCount`É o número de vezes que um consumidor pode receber uma mensagem de uma fila de origem antes de ser movida para uma fila de mensagens mortas. Por exemplo, se `maxReceiveCount` for definido como um valor baixo, como 1, uma falha no recebimento de uma mensagem faria com que a mensagem fosse movida para a fila de mensagens mortas. Para garantir que seu sistema seja resiliente contra erros, defina o `maxReceiveCount` alto o suficiente para permitir novas tentativas suficientes.

A política de permissão de redirecionamento especifica quais filas de origem podem acessar a fila de mensagens mortas. Você pode escolher se deseja permitir todas as filas de origem, permitir filas de origem específicas ou negar o uso da fila de mensagens mortas por todas as filas de origem. O padrão permite que todas as filas de origem usem a fila de mensagens mortas. Se você optar por permitir filas específicas usando a `byQueue` opção, poderá especificar até 10 filas de origem usando a fila de origem Amazon Resource Name (ARN). Se você especificar `denyAll`, a fila não pode ser usada como uma fila de mensagens mortas.

Entendendo os períodos de retenção de mensagens para filas de mensagens sem saída

Para filas comuns, a validade de uma mensagem é sempre baseada em seu carimbo de data/hora de enfileiramento original. Quando uma mensagem é movida para uma fila de mensagens mortas, o carimbo de data/hora de enfileiramento permanece inalterado. A `ApproximateAgeOfOldestMessage` métrica indica quando a mensagem foi movida para a fila de mensagens mortas, não quando a mensagem foi enviada originalmente. Por exemplo, suponha

que uma mensagem fique um dia na fila original antes de ser movida para uma fila de mensagens mortas. Se o período de retenção da fila de mensagens mortas for de quatro dias, a mensagem será excluída da fila de mensagens mortas após três dias e a `ApproximateAgeOfOldestMessage` será de três dias. Portanto, é uma prática recomendada definir sempre o período de retenção de uma fila de mensagens mortas para ser maior do que o período de retenção da fila original.

Para filas FIFO, o carimbo de data/hora da fila é redefinido quando a mensagem é movida para uma fila de mensagens não entregues. A métrica `ApproximateAgeOfOldestMessage` indica quando a mensagem foi movida para a fila de mensagens não entregues. No mesmo exemplo acima, a mensagem é excluída da fila de mensagens não entregues após quatro dias e `ApproximateAgeOfOldestMessage` é de quatro dias.

Saiba como configurar uma fila de mensagens sem saída usando o console do Amazon SQS

Uma fila de mensagens sem saída é uma fila que as filas de origem podem direcionar para mensagens que não foram processadas com êxito. Para ter mais informações, consulte Usando filas de mensagens mortas no Amazon SQS .

O Amazon SQS não cria fila de mensagens mortas automaticamente. Primeiro, é necessário criar uma fila antes de designar para ela uma fila de mensagens mortas. Para obter instruções sobre como criar uma fila para usar como fila de letras mortas, consulte. Crie uma fila usando o console do Amazon SQS

A fila de mensagens mortas de uma fila FIFO também deve ser uma fila FIFO. Da mesma forma, a fila de mensagens mortas de uma fila padrão também deve ser uma fila padrão.

Quando você cria ou edita uma fila, uma fila, você pode configurar uma fila de mensagens mortas.

Para configurar uma fila de mensagens mortas para uma fila existente (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Selecione uma fila e escolha Editar.
4. Role até a seção Dead-letter queue (Fila de mensagens mortas) e escolha Enabled (Habilitado).
5. Escolha o nome do recurso da Amazon (ARN) de uma fila de mensagens mortas que você deseja associar a essa fila de origem.

6. Para configurar o número de vezes que uma mensagem pode ser recebida antes de ser enviada a uma fila de mensagens mortas, defina Maximum receives (Recebimentos máximos) como um valor entre 1 e 1.000.
7. Quando você terminar de configurar a fila de mensagens mortas, escolha Save (Salvar).

   Depois de salvar a fila, o console exibe a página Details (Detalhes) de sua fila. Na página Details (Detalhes), a guia Dead-letter queue (Fila de mensagens mortas) exibe os Maximum Receives (Recebimentos máximos) e o ARN da fila de mensagens mortas em Dead-letter queue (Fila de mensagens mortas).

Saiba como configurar um redrive de fila de cartas mortas no Amazon SQS

Você pode usar o redirecionamento de fila de mensagens mortas para mover mensagens não consumidas de uma fila de mensagens não consumidas existente. Por padrão, a redirecionamento da fila de mensagens mortas move as mensagens de uma fila de mensagens mortas para uma fila de origem. No entanto, também será possível configurar qualquer outra fila como o destino de redirecionamento se as filas forem do mesmo tipo. Por exemplo, se a fila de mensagens não entregues for uma fila FIFO, a fila de destino de redirecionamento também deverá ser uma fila FIFO. Além disso, você pode configurar a velocidade de redirecionamento para definir a taxa na qual o Amazon SQS move mensagens.

> **Note**
>
> Quando uma mensagem é movida de uma fila FIFO para uma DLQ FIFO, a ID de desduplicação da mensagem original será substituída pela ID da mensagem original. Isso acontece para garantir que a eliminação de duplicação da DLQ não impeça o armazenamento de duas mensagens independentes que, por acaso, compartilham o mesmo ID de eliminação de duplicação.

As filas de mensagens mortas redirecionam as mensagens na ordem em que são recebidas, começando pela mensagem mais antiga. No entanto, a fila de destino ingere as mensagens redirecionadas, bem como as novas mensagens de outros produtores, de acordo com a ordem em que as recebe. Por exemplo, se um produtor estiver enviando mensagens para uma fila FIFO de origem ao receber simultaneamente mensagens redirecionadas de uma fila de letras mortas, as mensagens redirecionadas se entrelaçarão com as novas mensagens do produtor.

> **Note**
>
> A tarefa de redirecionamento redefine o período de retenção. Todas as mensagens redirecionadas são consideradas novas mensagens com uma nova `messageID` e `enqueueTime` são atribuídas a mensagens redirecionadas.

Tópicos

Configurando um redrive de fila de correio morto para uma fila padrão existente usando a API do Amazon SQS

Você pode configurar um redrive de fila de mensagens mortas usando as açõesSendMessageBatch,ReceiveMessage, e da API: DeleteMessageBatch

Ação API	Descrição
StartMessageMoveTask	Inicia uma tarefa assíncrona para mover mensagens de uma fila de origem especificada a uma fila de destino especificada.
ListMessageMoveTasks	Exibe as tarefas mais recentes de movimentação de mensagens (até dez) em uma fila de origem específica.
CancelMessageMoveTask	Cancela uma tarefa de movimentação de mensagens especificada. A movimentação de uma mensagem só pode ser cancelada quando o status atual é EM EXECUÇÃO.

Configurando um redrive de fila de correio morto para uma fila padrão existente usando o console do Amazon SQS

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Escolha o nome da fila configurada como uma fila de mensagens não entregues.
4. Selecione Start DLQ redrive (Iniciar o redirecionamento DLQ).
5. Em Redrive configuration (Configurações de redirecionamento), em Message destination (Destino da mensagem), execute uma das seguintes ações:
   - Para redirecionar mensagens para a fila de origem, escolha Redrive to source queue(s) (Redirecionar para fila(s) de origem).
   - Para redirecionar mensagens para outra fila, escolha Redrive to custom destination (Redirecionar para um destino personalizado). Em seguida, insira o nome do recurso da Amazon (ARN) de uma fila de destino existente.
6. Em Velocity control settings (Configurações de controle de velocidade), escolha uma das opções a seguir:
   - System optimized (Otimizado para o sistema) — Redirecione mensagens de fila de mensagens não entregues no número máximo de mensagens por segundo.
   - Custom max velocity (Velocidade máxima personalizada) — Redirecione mensagens de fila de mensagens não entregues com uma taxa máxima personalizada de mensagens por segundo. A taxa máxima permitida é de 500 mensagens por segundo.
     - É recomendável começar com um valor pequeno para a velocidade máxima personalizada e verificar se a fila de origem não está sobrecarregada com mensagens. A partir daí, aumente gradualmente o valor de velocidade máxima personalizada, continuando a monitorar o estado da fila de origem.
7. Quando você terminar de configurar o redirecionamento da fila de mensagens não entregues, escolha Save (Salvar).

> **Important**
>
> O Amazon SQS não oferece suporte à filtragem e modificação de mensagens enquanto as redireciona da fila de mensagens não entregues.

Uma tarefa de redirecionamento de fila de mensagens não entregues pode ser executada no máximo 36 horas. O Amazon SQS oferece suporte a um máximo de 100 tarefas de redirecionamento ativo por conta.

8. Se você quiser cancelar a tarefa de redirecionamento de mensagens, na página Details (Detalhes) da sua fila, escolha Cancel DLQ redrive (Cancelar redirecionamento DLQ). Ao cancelar uma redirecionamento de mensagem em andamento, todas as mensagens que já tenham sido movidas com sucesso para a fila de destino de movimentação permanecerão na fila de destino.

Configurar permissões de fila para o redirecionamento da fila de mensagens não entregues

Você pode conceder ao usuário acesso a ações específicas da fila de mensagens não entregues adicionando permissões à política. As permissões mínimas necessárias para uma fila de mensagens não entregues são as seguintes:

Permissões mínimas	Métodos de API necessários
Para iniciar um redirecionamento de mensagens	• Adicione `sqs:StartMessageMoveTask` , `sqs:ReceiveMessage` , `sqs:DeleteMessage` e `sqs:GetQueueAttributes` da fila de mensagens não entregues. Se a fila de mensagens não entregues ou a fila de origem forem criptografadas (também conhecida como fila SSE), também será necessário `kms:Decrypt` para qualquer chave do KMS usada para criptografar as mensagens. • Adicione o `sqs:SendMessage` da fila de destino. Se a fila de destino estiver criptografada, também será necessário adicionar `kms:GenerateDataKey` e `kms:Decrypt` .
Para cancelar um redirecionamento de mensagem em andamento	• Adicione `sqs:CancelMessageMoveTask` , `sqs:ReceiveMessage` , `sqs:DeleteMessage` e `sqs:GetQueueAttributes` da fila de mensagens não entregues. Se a fila de mensagens não entregues estiver

Permissões mínimas	Métodos de API necessários
	criptografada (também conhecida como fila SSE), também será necessário adicionar `kms:Decrypt` .
Para exibir o status de movimentação de uma mensagem	• Adicione `sqs:ListMessageMoveTasks` e `sqs:GetQueueAttributes` da fila de mensagens não entregues.

Para configurar permissões a um par de filas criptografadas (uma fila de origem com uma fila de mensagens não entregues)

Siga as etapas abaixo para configurar permissões mínimas de redirecionamento de fila de mensagens não entregues:

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies.
3. Crie uma política com as seguintes permissões e anexe-a ao usuário ou à função de login do IAM:
   - `sqs:StartMessageMoveTask`
   - `sqs:CancelMessageMoveTask`
   - `sqs:ListMessageMoveTasks`
   - `sqs:ListDeadLetterSourceQueues`
   - `sqs:ReceiveMessage`
   - `sqs:DeleteMessage`
   - `sqs:GetQueueAttributes`
   - O ARN de `Resource` da fila de mensagens não entregues (por exemplo, “arn:aws:sqs:*<região_da_fila>*:*<ID_da_conta_da_fila>*:*<nome_da_fila>*”).
   - `sqs:SendMessage`
   - O `Resource` ARN da fila de destino (por exemplo, “arn:aws:sqs: < _region>: < *_accountId>: < DestQueue _name>* “). DestQueue DestQueue

- `kms:Decrypt`: permite a ação de descriptografia.
- `kms:GenerateDataKey`
- Os ARNs de `Resource` de qualquer chave de criptografia do KMS usada para criptografar as mensagens na fila de origem (por exemplo, “arn:aws:kms:*<region>*:*<accountId>*:key/*<keyId_usada para criptografar o corpo da mensagem>*”).
- O ARN da chave de criptografia do KMS usada para a fila de destino de redirecionamento (por exemplo, “arn:aws:kms:*<region>*:*<accountId>*:key/*<keyId_usada para a fila de destino>*”).

A política de acesso deve ser semelhante a:

```
{
  "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>"
    }
  ]
}
```

Para configurar permissões a um par de filas não criptografadas (uma fila de origem com uma fila de mensagens não entregues)

Siga as etapas abaixo para configurar permissões mínimas para uma fila de mensagens não entregues padrão não criptografadas. As permissões mínimas necessárias são para receber, excluir e obter atributos da fila de mensagens não entregues e enviar atributos para a fila de origem.

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies.
3. Crie uma política com as seguintes permissões e anexe-a ao usuário ou à função de login do IAM:

   - `sqs:StartMessageMoveTask`
   - `sqs:CancelMessageMoveTask`
   - `sqs:ListMessageMoveTasks`
   - `sqs:ListDeadLetterSourceQueues`
   - `sqs:ReceiveMessage`
   - `sqs:DeleteMessage`
   - `sqs:GetQueueAttributes`
   - O ARN de `Resource` da fila de mensagens não entregues (por exemplo, “arn:aws:sqs:*<região_da_fila>*:*<ID_da_conta_da_fila>*:*<nome_da_fila>*”).
   - `sqs:SendMessage`
   - *O Resource ARN da fila de destino (por exemplo, “arn:aws:sqs: < _region>: < _accountId>: < DestQueue _name> “). DestQueue DestQueue*

   A política de acesso deve ser semelhante a:

```
{
"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 atualização e permissão para o redrive de fila de cartas mortas do Amazon SQS

Em 8 de junho de 2023, o Amazon SQS introduziu o redrive de fila de letras mortas (DLQ) para AWS SDK e (CLI). AWS Command Line Interface Esse recurso é uma adição ao redrive DLQ já suportado para o AWS console. Se você já usou o AWS console para redirecionar mensagens da fila de mensagens mortas, você pode ser afetado pelas seguintes alterações:

- CloudTrail renomeação de eventos para redirecionamento de fila de mensagens mortas
- Atualização das permissões de redirecionamento da fila de mensagens não entregues

CloudTrail renomeação de eventos

Em 15 de outubro de 2023, os nomes dos CloudTrail eventos para recondução de filas de cartas mortas serão alterados no console do Amazon SQS. Se você definiu alarmes para esses CloudTrail

eventos, você deve atualizá-los agora. A seguir estão os novos nomes de CloudTrail eventos para o DLQ redrive:

Antigo nome do evento	Novo nome do evento
`CreateMoveTask`	`StartMessageMoveTask`
`CancelMoveTask`	`CancelMessageMoveTask`

Permissões atualizadas

Incluído na versão do SDK e da CLI, o Amazon SQS também atualizou as permissões de fila para o redirecionamento da DLQ a fim de aderir às práticas recomendadas de segurança. Use os tipos de permissão de fila a seguir para redirecionar mensagens das DLQs.

1. Permissões baseadas em ações (atualização para as ações da API da DLQ)
2. Permissões de políticas gerenciadas do Amazon SQS
3. Política de permissão que usa o curinga sqs:*

> **Important**
>
> Para usar o redirecionamento de DLQ para o SDK ou a CLI, é necessário ter uma política de permissão de redirecionamento de DLQ que corresponda a uma das opções acima.

Se as permissões de fila para o redirecionamento da DLQ não corresponderem a uma das opções acima, você deverá atualizá-las até 31 de agosto de 2023. Até 31 de agosto de 2023, sua conta poderá redirecionar mensagens usando as permissões que você configurou por meio do console da AWS apenas nas regiões em que você já usou o redirecionamento de DLQ. Por exemplo, digamos que você tenha a “Conta A” em us-east-1 e eu-west-1. A “Conta A” foi usada para redirecionar mensagens no AWS console em us-east-1 antes de 8 de junho de 2023, mas não em eu-west-1. Entre 8 de junho de 2023 e 31 de agosto de 2023, se as permissões da política da “Conta A” não corresponderem a uma das opções acima, elas só poderão ser usadas para redirecionar mensagens no AWS console em us-east-1, e não em eu-west-1.

> **Important**
>
> Se as permissões de redirecionamento da DLQ não corresponderem a uma dessas opções após 31 de agosto de 2023, sua conta não poderá mais redirecionar mensagens da DLQ usando o console da AWS .
> No entanto, se você usou o recurso de redrive DLQ no AWS console em agosto de 2023, terá uma extensão até 15 de outubro de 2023 para adotar as novas permissões de acordo com uma dessas opções.
> Para ter mais informações, consulte the section called “Identificação de políticas afetadas”.

Veja a seguir exemplos de permissões de fila para cada opção de redirecionamento de DLQ. Ao usar filas criptografadas do lado do servidor (SSE), a permissão de AWS KMS chave correspondente é necessária.

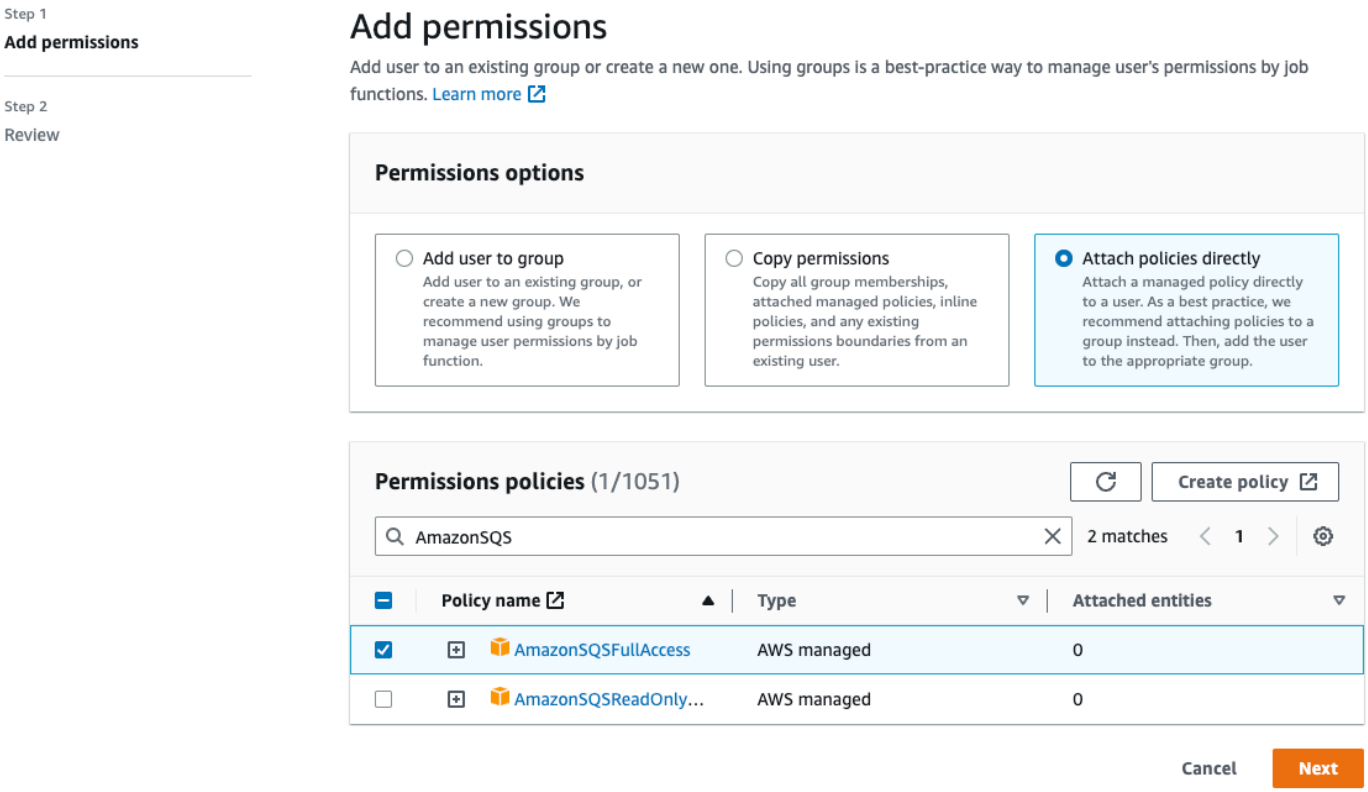
Baseado em ação

```
{
  "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 gerenciada

As seguintes políticas gerenciadas contém as permissões atualizadas exigidas:

- AmazonSQS FullAccess — Inclui as seguintes tarefas de recondução de filas sem saída: iniciar, cancelar e listar.
- Acesso ao AmazonSQS — Fornece ReadOnly acesso somente para leitura e inclui a tarefa de recondução da fila de cartas mortas da lista.



Política de permissão que usa o curinga sqs*

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

```
}
```

Identificação de políticas afetadas

Se você estiver usando políticas gerenciadas pelo cliente (CMPs), você pode usar AWS CloudTrail um IAM para identificar as políticas afetadas pela atualização de permissões de fila.

> **Note**
>
> Se você estiver usando `AmazonSQSFullAccess` e `AmazonSQSReadOnlyAccess`, nenhuma ação será necessária.

1. Faça login no AWS CloudTrail console.
2. Na página Histórico de eventos, em Pesquisar atributos, use o menu suspenso para selecionar Nome do evento. Depois, pesquise `CreateMoveTask`.
3. Escolha um evento para abrir a página Detalhes. Na seção Registros de eventos, recupere o `UserName` ou `RoleName` do ARN de `userIdentity`.
4. Faça login no console do IAM.
   - Para usuários, escolha “Usuários”. Selecione o usuário com o `UserName` identificado na etapa anterior.
   - Para perfis, escolha “Perfis”. Pesquise o usuário com o `RoleName` identificado na etapa anterior.
5. No página Detalhes, na seção Permissões, revise todas as políticas com o prefixo `sqs:` em `Action`, ou revise as políticas que tenham a fila do Amazon SQS definida em `Resource`.

Crie alarmes para filas de mensagens sem saída usando a Amazon CloudWatch

Você pode configurar um alarme para qualquer mensagem movida para uma fila de mensagens mortas usando a Amazon CloudWatch e a métrica. `ApproximateNumberOfMessagesVisible` Para ter mais informações, consulte Criação de CloudWatch alarmes para métricas do Amazon SQS. Depois de receber um alerta de que as mensagens foram enviadas para a fila de mensagens mortas, você pode revisar as mensagens usando a enquete para receber a mensagem.

Metadados de mensagens para o Amazon SQS

É possível usar atributos de mensagem para associar metadados personalizados a mensagens do Amazon SQS para suas aplicações. É possível usar atributos do sistema de mensagens a fim de armazenar metadados para outros serviços da AWS , como o AWS X-Ray.

Tópicos

Atributos de mensagem do Amazon SQS

O Amazon SQS permite que você inclua metadados estruturados (como carimbos de data e hora, dados geoespaciais, assinaturas e identificadores) com mensagens usando os atributos de mensagem. Cada mensagem pode ter até dez atributos. Os atributos de mensagem são opcionais e separados do corpo da mensagem (no entanto, são enviados junto com o corpo da mensagem). O consumidor pode usar atributos de mensagem para tratar uma mensagem de uma forma específica sem precisar primeiro processar o corpo da mensagem. Para obter informações sobre como enviar mensagens com atributos usando o console do Amazon SQS, consulte Enviar uma mensagem com atributos.

> **Note**
>
> Não confunda atributos de mensagem com atributos do sistema de mensagens: embora você possa usar atributos de mensagem para anexar metadados personalizados às mensagens do Amazon SQS para seus aplicativos, você pode usar atributos do sistema de mensagens para armazenar metadados para AWS outros serviços, como. AWS X-Ray

Tópicos

Componentes de atributos de mensagem

> **Important**
>
> Todos os componentes de um atributo de mensagem estão incluídos na restrição de tamanho de 256 KB da mensagem.
> O `Name`, `Type`, `Value` e o corpo da mensagem não devem estar vazios ou serem nulos.

Cada atributo de mensagem consiste nos seguintes componentes:

- Nome: o nome do atributo da mensagem pode conter os seguintes caracteres: `A-Z`, `a-z`, `0-9`, sublinhado (`_`), hífen (`-`), e ponto (`.`). As seguintes restrições são aplicáveis:
  - Pode ter até 256 caracteres
  - Não pode começar com `AWS.` ou `Amazon.` (ou qualquer variação no uso de maiúsculas e minúsculas)
  - Diferencia maiúsculas de minúsculas
  - Deve ser exclusivo entre todos os nomes de atributos da mensagem
  - Não deve começar ou terminar com um ponto
  - Não deve ter pontos em uma sequência
- Tipo: o tipo de dados do atributo da mensagem. Os tipos compatíveis incluem `String`, `Number` e `Binary`. Você também pode adicionar informações personalizadas para qualquer tipo de dados. O tipo de dados tem as mesmas restrições que o corpo da mensagem (para obter mais informações, consulte `SendMessage` na Referência da API do Amazon Simple Queue Service). Além disso, aplicam-se as seguintes restrições:
  - Pode ter até 256 caracteres
  - Diferencia maiúsculas de minúsculas
- Valor: o valor do atributo da mensagem. Para tipos de dados `String`, os valores dos atributos têm as mesmas restrições que o corpo da mensagem.

Tipos de dados de atributos de mensagem

Os tipos de dados de atributos de mensagens indicam ao Amazon SQS como tratar valores de atributos de mensagens correspondentes. Por exemplo, se o tipo for `Number`, o Amazon SQS validará valores numéricos.

O Amazon SQS é compatível com os tipos de dados lógicos `String`, `Number` e `Binary` com rótulos de tipos de dados personalizados opcionais com o formato *`.custom-data-type`*

- String: os atributos `String` podem armazenar texto Unicode usando quaisquer caracteres XML válidos.
- Número: os atributos `Number` podem armazenar valores numéricos positivos ou negativos. Um número pode ter até 38 dígitos de precisão, e pode ser entre 10^-128 e 10^+126.

> **Note**
>
> O Amazon SQS remove zeros iniciais e finais.

- Binário: os atributos binários podem armazenar qualquer dado binário, como dados compactados, dados criptografados ou imagens.
- Personalizado: para criar um tipo de dado personalizado, acrescente um rótulo de tipo personalizado a qualquer tipo de dado. Por exemplo: .
  - `Number.byte`, `Number.short`, `Number.int` e `Number.float` podem ajudar a diferenciar entre tipos numéricos.
  - `Binary.gif` e `Binary.png` podem ajudar a diferenciar entre tipos de arquivos.

> **Note**
>
> O Amazon SQS não interpreta, valida ou usa os dados anexados.
> O rótulo de tipo personalizado tem as mesmas restrições que o corpo da mensagem.

Cálculo do resumo de mensagens MD5 para atributos de mensagem

Se você usar o AWS SDK for Java, você pode pular esta seção. A classe `MessageMD5ChecksumHandler` do SDK for Java oferece suporte a resumos de mensagens MD5 para atributos de mensagens do Amazon SQS.

Se você usar a API de consulta ou um dos AWS SDKs que não suporta resumos de mensagens MD5 para atributos de mensagens do Amazon SQS, você deve usar as diretrizes a seguir para realizar o cálculo do resumo de mensagens MD5.

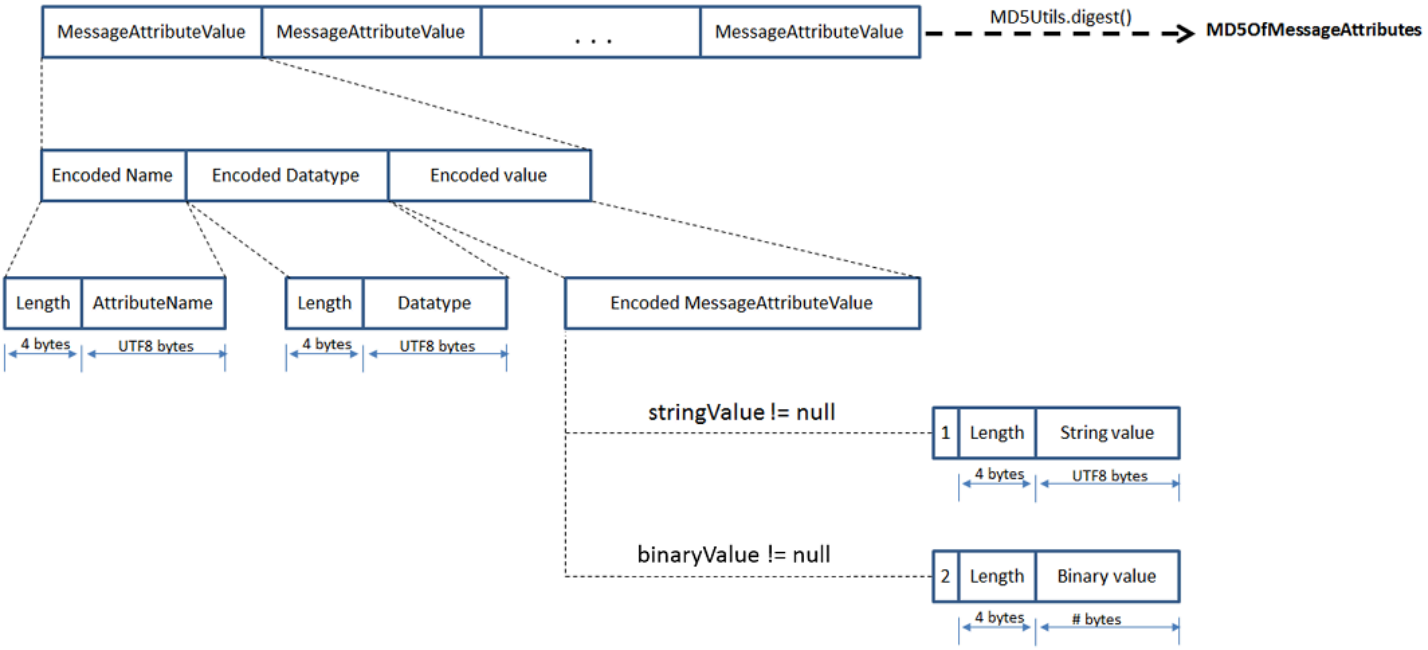
> **Note**
>
> Sempre inclua sufixos de tipos de dados personalizados no cálculo do resumo de mensagens MD5.

Visão geral

O seguinte é uma visão geral do algoritmo de cálculo do resumo de mensagens MD5:

1. Classificar todos os atributos de mensagem por nome em ordem crescente.
2. Codificar as partes individuais de cada atributo (`Name`, `Type` e `Value`) em um buffer.
3. Calcular o resumo de mensagem de todo o buffer.

O seguinte diagrama mostra a codificação do resumo de mensagens MD5 para um único atributo de mensagem:



Para codificar um único atributo de mensagem do Amazon SQS

1. Codifique o nome: o comprimento (4 bytes) e os bytes UTF-8 do nome.
2. Codifique o tipo de dados: o comprimento (4 bytes) e os bytes UTF-8 do tipo de dados.
3. Codifique o tipo de transporte (`String` ou `Binary`) do valor (1 byte).

> **Note**
>
> Os tipos de dados lógicos `String` e `Number` usam o tipo de transporte `String`.
> Os tipos de dados lógicos `Binary` usam o tipo de transporte `Binary`.

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

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

4. Codifique o valor do atributo.

   a. Para o tipo de transporte `String`, codifique o valor do atributo: o comprimento (4 bytes) e os bytes UTF-8 do valor.

   b. Para o tipo de transporte `Binary`, codifique o valor do atributo: o comprimento (4 bytes) e os bytes brutos do valor.

Atributos do sistema de mensagens do Amazon SQS

Enquanto é possível usar atributos de mensagens para anexar metadados personalizados a mensagens do Amazon SQS para suas aplicações, é possível usar atributos do sistema de mensagens a fim de armazenar metadados para outros produtos da AWS , como o AWS X-Ray. Para obter mais informações, consulte o parâmetro de solicitação `MessageSystemAttribute` das ações de API de `SendMessage` e `SendMessageBatch`, o atributo `AWSTraceHeader` da ação de API `ReceiveMessage` e o tipo de dado `MessageSystemAttributeValue` na Referência da API do Amazon Simple Queue Service.

Os atributos do sistema de mensagens são estruturados exatamente como atributos de mensagens, com as seguintes exceções:

- Atualmente, o único atributo do sistema de mensagens compatível é `AWSTraceHeader`. Seu tipo deve ser `String` e seu valor deve ser uma string de cabeçalho de AWS X-Ray rastreamento formatada corretamente.
- O tamanho de um atributo do sistema de mensagens não entra na contagem para o tamanho total de uma mensagem.

Recursos necessários para processar mensagens do Amazon SQS

Para ajudar você a estimar os recursos necessários para processar mensagens na fila, o Amazon SQS pode determinar a quantidade aproximada de mensagens em atraso, visíveis e não visíveis em uma fila. Para obter mais informações sobre visibilidade, consulte Tempo limite de visibilidade do Amazon SQS.

> **Note**
>
> Para filas padrão, o resultado é aproximado por causa da arquitetura distribuída do Amazon SQS. Na maioria dos casos, a contagem deve ser próxima da quantidade real de mensagens na fila.
> Para filas FIFO, o resultado é exato.

A tabela a seguir lista o nome do atributo a ser usado com a ação `GetQueueAttributes`:

Tarefa	Nome do atributo
Obter o número de mensagens disponíveis para recuperação na fila.	`ApproximateNumberOfMessages Visible`
Obter o número de mensagens na fila que estão atrasadas e indisponíveis para leitura imediata. Isso pode acontecer quando a fila tem a configuração de fila com atraso ou quando uma mensagem foi enviada com um parâmetro de atraso.	`ApproximateNumberOfMessages Delayed`
Obter o número de mensagens que estão em processamento. As mensagens são considera das como em processamento quando foram enviadas a um cliente, mas ainda não foram excluídas ou ainda não atingiram o final de sua janela de visibilidade.	`ApproximateNumberOfMessages NotVisible`

Listar paginação de filas

Os métodos de API `listQueues` e `listDeadLetterQueues` oferecem suporte a controles opcionais de paginação. Por padrão, esses métodos de API retornam até 1.000 filas na mensagem de resposta. É possível definir o parâmetro `MaxResults` para retornar menos resultados em cada resposta.

Defina o parâmetro `MaxResults` na solicitação `listQueues` ou `listDeadLetterQueues` para especificar o número máximo de resultados a serem retornados na resposta. Se você não definir o `MaxResults`, a resposta incluirá um máximo de 1.000 resultados e o valor de `NextToken` na resposta será nulo.

Se você definir `MaxResults`, a resposta incluirá um valor para `NextToken`, se houver resultados adicionais a serem exibidos. Use `NextToken` como parâmetro na próxima solicitação para `listQueues` a fim de receber a próxima página de resultados. Se não houver resultados adicionais a serem exibidos, o valor de `NextToken` na resposta será nulo.

Tags de alocação de custos do Amazon SQS

Para organizar e identificar as filas do Amazon SQS para alocação de custos, você pode adicionar tags de metadados que identificam o proprietário, o ambiente ou a finalidade de uma fila. Isso é especialmente útil quando você tem várias filas. Para configurar tags usando o console do Amazon SQS, consulte the section called “Configurar tags para uma fila”

Você pode usar etiquetas de alocação de custos para organizar sua AWS fatura de forma a refletir sua própria estrutura de custos. Para fazer isso, inscreva-se para que sua Conta da AWS fatura inclua as chaves e valores das etiquetas. Para obter mais informações, consulte Configuração de um relatório de alocação de custos mensal no Guia do usuário do AWS Billing .

Toda tag é composta de um par de valores de chave, que são definidos por você. Por exemplo, você pode identificar facilmente suas filas de produção e teste se você marcar suas filas da seguinte forma:

Fila	Chave	Valor
`MyQueueA`	`QueueType`	`Production`
`MyQueueB`	`QueueType`	`Testing`

> **Note**
>
> Ao usar tags de fila, lembre-se das orientações a seguir:
>
> - Não recomendamos adicionar mais de 50 tags a uma fila. A marcação é compatível com caracteres Unicode em UTF-8.
> - As tags não têm significado semântico. O Amazon SQS interpreta as tags como strings.
> - As tags diferenciam letras maiúsculas de minúsculas.
> - Uma nova tag com uma chave idêntica à de uma tag existente substitui a tag existente.
> - As ações de marcação são limitadas a 30 TPS por. Conta da AWS Se a sua aplicação exigir um throughput mais alto, envie uma solicitação.
>
> Para ver uma lista completa de restrições de etiquetas, consulte Cotas.

Sondagem curta e longa do Amazon SQS

O Amazon SQS oferece opções de sondagem curta e longa para receber mensagens de uma fila. Considere os requisitos de capacidade de resposta e eficiência de custos do seu aplicativo ao escolher entre essas duas opções de pesquisa:

- Pesquisa curta (padrão) — A `ReceiveMessage`solicitação consulta um subconjunto de servidores (com base em uma distribuição aleatória ponderada) para encontrar as mensagens disponíveis e envia uma resposta imediata, mesmo que nenhuma mensagem seja encontrada.
- Pesquisa longa — `ReceiveMessage`consulta mensagens em todos os servidores, enviando uma resposta quando pelo menos uma mensagem estiver disponível, até o máximo especificado. Uma resposta vazia será enviada somente se o tempo de espera da pesquisa expirar. Essa opção pode reduzir o número de respostas vazias e potencialmente reduzir os custos.

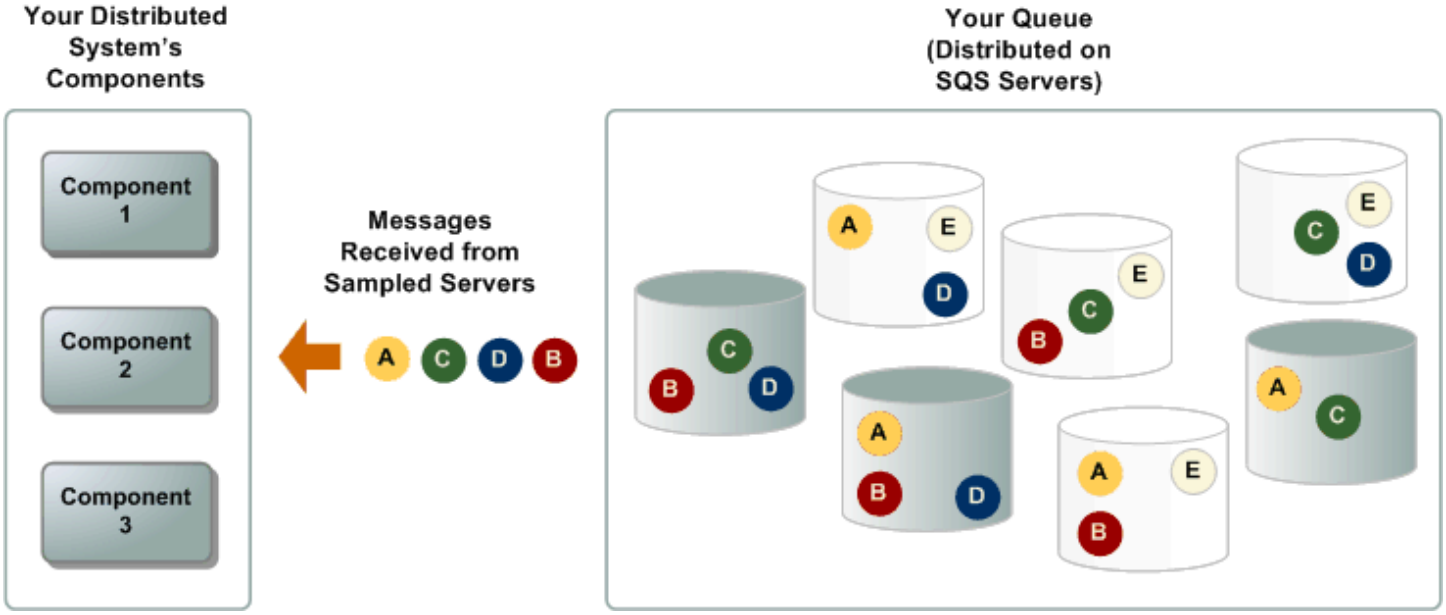
As seções a seguir explicam os detalhes de sondagens curtas e sondagens longas.

Tópicos

Consumo de mensagens usando sondagem curta

Quando você consome mensagens de uma fila (FIFO ou padrão) usando uma sondagem curta, o Amazon SQS coleta amostras de um subconjunto de seus servidores (com base em uma distribuição aleatória ponderada) e retorna mensagens somente desses servidores. Assim, uma determinada solicitação `ReceiveMessage` pode não retornar todas as suas mensagens. No entanto, se você tiver menos de 1.000 mensagens na fila, uma solicitação subsequente retornará suas mensagens. Se você continuar consumindo em suas filas, o Amazon SQS obterá amostras de todos os seus servidores, e você receberá todas as mensagens.

O diagrama a seguir mostra o comportamento da sondagem curta de mensagens retornadas de uma fila padrão depois que um dos componentes do sistema faz uma solicitação de recebimento. O Amazon SQS analisa vários dos seus servidores (em cinza) e retorna as mensagens A, C, D e B desses servidores. A mensagem E não é retornada para essa solicitação, mas é retornada para uma solicitação subsequente.



Consumo de mensagens usando a sondagem longa

Quando o tempo de espera da ação da API `ReceiveMessage` é maior do que 0, a sondagem longa está em vigor. O tempo máximo de espera de sondagem longa é de 20 segundos. A sondagem longa ajuda a reduzir os custos de uso do Amazon SQS eliminando o número de respostas vazias (quando não há mensagens disponíveis para uma solicitação `ReceiveMessage`) e respostas vazias falsas (quando mensagens estão disponíveis, mas não são incluídas em uma resposta). Para obter informações sobre como habilitar a sondagem longa para uma fila nova ou existente usando o

console do Amazon SQS, consulte Configurando parâmetros de fila usando o console do Amazon SQS. Para ver as práticas recomendadas, consulte Configuração da sondagem longa.

A sondagem longa oferece os seguintes benefícios:

- Reduza as respostas vazias permitindo que o Amazon SQS espere até que uma mensagem esteja disponível em uma fila antes de enviar uma resposta. A menos que uma conexão expire, a resposta à solicitação `ReceiveMessage` contém pelo menos uma das mensagens disponíveis, até o número máximo de mensagens especificado na ação `ReceiveMessage`. Em casos raros, você pode receber respostas vazias mesmo quando uma fila ainda contiver mensagens, especialmente se você especificar um valor baixo para o parâmetro `ReceiveMessageWaitTimeSeconds`.
- Reduza respostas vazias falsas consultando todos os servidores do Amazon SQS, não apenas um subconjunto deles.
- Retornar mensagens assim que se tornam disponíveis.

Para obter mais informações sobre como confirmar se uma fila está vazia, consulte Confirmando que uma fila do Amazon SQS está vazia.

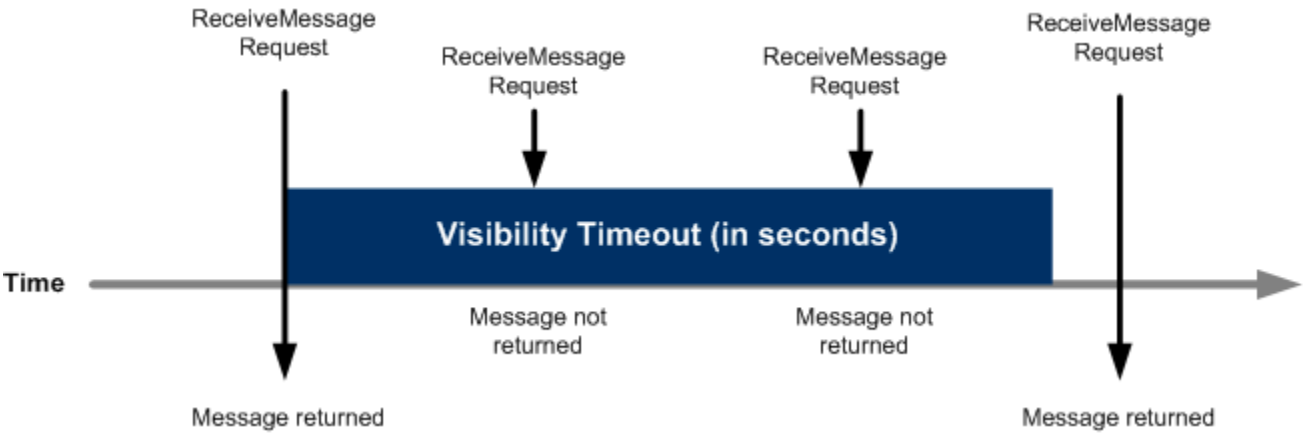
Diferenças entre as sondagens longa e curta

A sondagem curta ocorre quando o parâmetro `WaitTimeSeconds` de uma solicitação `ReceiveMessage` é definido como `0` de uma de duas maneiras:

- A chamada `ReceiveMessage` define `WaitTimeSeconds` como `0`.
- A chamada `ReceiveMessage` não define `WaitTimeSeconds`, mas o atributo da fila `ReceiveMessageWaitTimeSeconds` é definido como `0`.

Tempo limite de visibilidade do Amazon SQS

Quando um consumidor recebe e processa uma mensagem de uma fila, ela permanece na fila. O Amazon SQS não exclui a mensagem automaticamente. Como o Amazon SQS é um sistema distribuído, não há garantia de que o consumidor realmente receba a mensagem (por exemplo, por causa de um problema de conectividade ou um problema na aplicação consumidora). Desse modo, o consumidor deve excluir a mensagem da fila após o recebimento e processamento.



Logo após uma mensagem ser recebida, ela permanece na fila. Para evitar que outros consumidores processem a mensagem novamente, o Amazon SQS define um tempo limite de visibilidade, um período durante o qual o Amazon SQS impede que todos os consumidores recebam e processem a mensagem. O tempo limite de visibilidade padrão para uma mensagem é de 30 segundos. O mínimo é 0 segundo. O máximo é 12 horas. Para obter informações sobre como configurar o tempo limite de visibilidade para uma fila usando o console, consulte Configurando parâmetros de fila usando o console do Amazon SQS.

> **Note**
>
> Para filas padrão, o tempo limite de visibilidade não é uma garantia contra o recebimento de uma mensagem duas vezes. Para ter mais informações, consulte Uma t-least-once entrega. As filas FIFO permitem várias tentativas ao produtor ou ao consumidor:
>
> - Se o produtor detectar uma falha de ação `SendMessage`, ele poderá tentar enviá-la novamente quantas vezes for necessário, usando o mesmo ID de eliminação de duplicação de mensagens. Supondo que o produtor receba pelo menos uma confirmação antes que o intervalo de eliminação de duplicação expire, as várias tentativas não afetarão a ordenação das mensagens nem introduzirão duplicatas.
> - Se o consumidor detectar uma falha de ação `ReceiveMessage`, ele poderá tentar enviá-la novamente quantas vezes for necessário, usando o mesmo ID de tentativa de solicitação de recebimento. Supondo que o consumidor receba pelo menos uma confirmação antes do tempo limite de visibilidade expirar, as várias tentativas não afetarão a ordenação das mensagens.

- Quando você recebe uma mensagem com um ID de grupo de mensagens, nenhuma outra mensagem para o mesmo ID de grupo de mensagens são retornadas, a menos que você exclua a mensagem ou ela se torne visível.

Tópicos

Mensagens em trânsito

Uma mensagem do Amazon SQS tem três estados básicos:

1. Enviada para uma fila por um produtor.
2. Recebida da fila por um consumidor.
3. Excluída da fila.

Uma mensagem é considerada armazenada depois de ser enviada para uma fila por um produtor, mas enquanto ainda não é recebida da fila por um consumidor (isto é, entre os estados 1 e 2). Não há cota para o número de mensagens armazenadas. Uma mensagem é considerada em trânsito depois de ser enviada para uma fila por um produtor, mas enquanto ainda não é excluída da fila (isto é, entre os estados 2 e 3). Há uma cota para o número de mensagens em trânsito.

Important

As cotas que se aplicam a mensagens em trânsito não estão relacionadas ao número ilimitado de mensagens armazenadas.

Para a maioria das filas padrão (dependendo do tráfego da fila e da lista de pendências), pode haver um máximo de aproximadamente 120 mil mensagens em trânsito (recebidas de uma fila por um consumidor, mas ainda não excluídas da fila). Se você atingir essa cota ao usar a sondagem curta, o Amazon SQS retornará a mensagem de erro `OverLimit`. Se você usar a sondagem longa, o

Amazon SQS não retornará nenhuma mensagem de erro. Para evitar atingir o quota, você deve excluir mensagens da fila depois de serem processadas. Você também pode aumentar o número de filas que usar para processar as mensagens. Para solicitar um aumento, envie um pedido de suporte.

Para filas FIFO, pode haver no máximo de 20.000 mensagens em trânsito (recebidas de uma fila por um consumidor, mas ainda não excluídas da fila). Se você atingir essa cota, o Amazon SQS não retornará nenhuma mensagem de erro.

> **Important**
>
> Ao trabalhar com filas FIFO, ocorrerá uma falha nas operações `DeleteMessage` se a solicitação for recebida fora da janela de tempo limite de visibilidade. Se o tempo limite de visibilidade for 0 segundos, a mensagem deverá ser excluída dentro do mesmo milissegundo em que foi enviada ou considerada abandonada. Isso pode fazer com que o Amazon SQS inclua mensagens duplicadas na mesma resposta a uma operação `ReceiveMessage` se o parâmetro `MaxNumberOfMessages` for maior que 1. Para obter detalhes adicionais, consulte Como funciona a API FIFO do Amazon SQS.

Definição do tempo limite de visibilidade

O tempo limite de visibilidade começa quando o Amazon SQS retorna uma mensagem. Durante esse período, o consumidor processa e exclui a mensagem. No entanto, se o consumidor falhar antes de excluir a mensagem e seu sistema não chamar a ação `DeleteMessage` para essa mensagem antes que o tempo limite de visibilidade expire, a mensagem ficará visível para outros consumidores e a mensagem será recebida novamente. Se uma mensagem só deve ser recebida uma vez, o consumidor deverá excluí-la durante o tempo limite de visibilidade.

Toda fila do Amazon SQS tem a configuração padrão de 30 segundos para o tempo limite de visibilidade. Você pode alterar essa configuração para toda a fila. Normalmente, você deve definir o tempo limite de visibilidade como o tempo máximo de que o seu aplicativo precisa para processar e excluir uma mensagem da fila. Ao receber mensagens, você também pode definir um tempo limite de visibilidade especial para as mensagens retornadas sem alterar o tempo limite de fila geral. Para obter mais informações, consulte as melhores práticas na seção Processar mensagens em tempo hábil.

Se você não souber quanto tempo leva para processar uma mensagem, crie uma pulsação para o processo do consumidor: especifique o tempo limite de visibilidade inicial (por exemplo, 2 minutos)

e, desde que o consumidor ainda funcione na mensagem, continue estendendo o tempo limite de visibilidade em 2 minutos a cada minuto.

> **Important**
>
> O tempo limite de visibilidade máximo é de 12 horas a partir do momento em que o Amazon SQS recebe `ReceiveMessage`. Estender o tempo limite de visibilidade não redefine o período máximo de 12 horas.
> Além disso, talvez você não consiga definir o tempo limite em uma mensagem individual para as 12 horas completas (por exemplo, 43.200 segundos), pois a solicitação de `ReceiveMessage` inicia o temporizador. Por exemplo, se você receber uma mensagem e definir imediatamente o máximo de 12 horas enviando uma chamada de `ChangeMessageVisibility` com `VisibilityTimeout` igual a 43.200 segundos, provavelmente ocorrerá uma falha. No entanto, usar um valor de 43.195 segundos funcionará, a menos que haja um atraso significativo entre a solicitação da mensagem via `ReceiveMessage` e a atualização do tempo limite de visibilidade. Se o consumidor precisar de mais de 12 horas, considere usar o Step Functions.

Alteração do tempo limite de visibilidade de uma mensagem

Quando você recebe uma mensagem de uma fila e começa a processá-la, o tempo limite de visibilidade para a fila pode ser insuficiente (por exemplo, você pode precisar processar e excluir uma mensagem). Você pode reduzir ou estender a visibilidade da mensagem especificando um novo valor de tempo limite usando a ação `ChangeMessageVisibility`.

Por exemplo, se o tempo limite padrão de uma fila é 60 segundos, 15 segundos tiverem decorrido desde que você recebeu a mensagem, e você enviar uma chamada `ChangeMessageVisibility` com `VisibilityTimeout` definido como 10 segundos, os 10 segundos começam a contar a partir do momento em que você faz a chamada `ChangeMessageVisibility`. Portanto, qualquer tentativa de alterar o tempo limite de visibilidade ou excluir essa mensagem 10 segundos após inicialmente alterar o tempo limite de visibilidade (um total de 25 segundos) pode resultar em um erro.

> **Note**
>
> O novo tempo limite entra em vigor quando você chama a ação `ChangeMessageVisibility`. Além disso, o novo tempo limite se aplica apenas ao

recebimento específico da mensagem. `ChangeMessageVisibility` não afeta o tempo limite de recebimentos posteriores da mensagem ou de filas posteriores.

Término do tempo limite de visibilidade de uma mensagem

Quando você recebe uma mensagem de uma fila, pode descobrir que realmente não quer processar e excluir essa mensagem. O Amazon SQS permite que você termine o tempo limite de visibilidade para uma mensagem específica. Isso torna a mensagem imediatamente visível para outros componentes no sistema e disponível para processamento.

Para terminar o tempo limite de visibilidade de uma mensagem depois de chamar `ReceiveMessage`, chame `ChangeMessageVisibility` com `VisibilityTimeout` definido como 0 segundos.

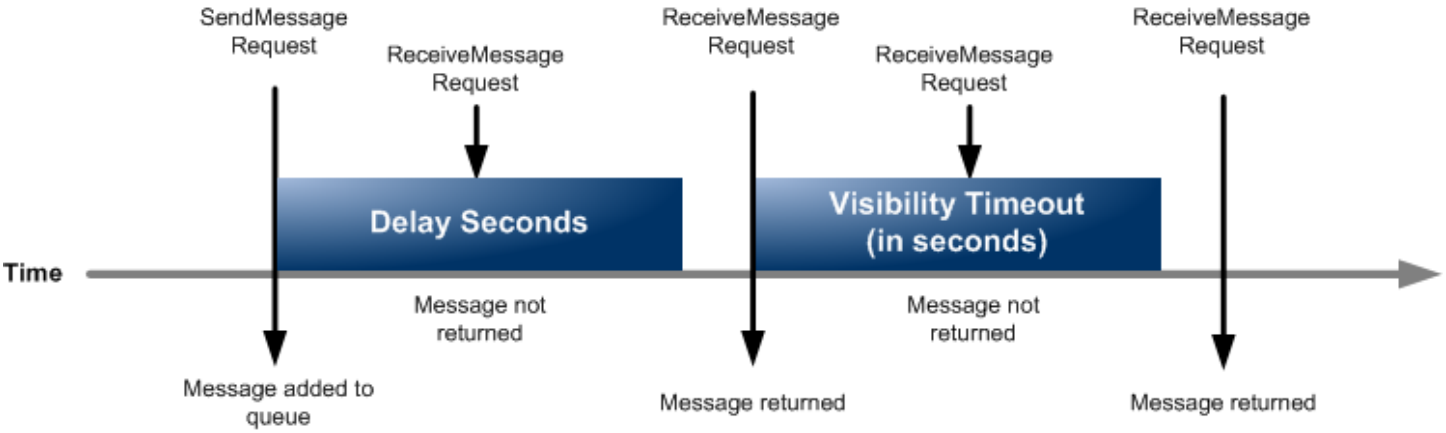
Filas de atraso do Amazon SQS

As filas de atraso permitem adiar a entrega de novas mensagens para consumidores por alguns segundos, por exemplo, quando sua aplicação de consumo precisa de tempo adicional para processar mensagens. Se você criar uma fila de atraso, qualquer mensagem enviada para essa fila permanecerá invisível para os consumidores durante o período de atraso. O atraso padrão (mínimo) para uma fila é 0 segundo. O máximo é 15 minutos. Para obter mais informações sobre como configurar filas de atraso usando o console, consulte Configurando parâmetros de fila usando o console do Amazon SQS.

> **Note**
>
> Para filas padrão, a configuração de atraso por fila não é retroativa. A alteração da configuração não afeta o atraso de mensagens que já estão na fila.
> Para filas FIFO, a configuração de atraso por fila é retroativa. A alteração da configuração afeta o atraso de mensagens que já estão na fila.

As filas de atraso são semelhantes a tempos limite de visibilidade, pois os dois recursos tornam as mensagens indisponíveis para os consumidores por um período específico. A diferença entre os dois é que, para filas de atraso, uma mensagem é ocultada quando é adicionada à fila pela primeira vez, enquanto que para os tempos limite de visibilidade uma mensagem é ocultada somente depois que

a mensagem é consumida na fila. O diagrama a seguir ilustra a relação entre filas de atraso e os tempos limite de visibilidade.



Para definir os segundos de atraso em mensagens individuais, em vez de em uma fila inteira, use temporizadores de mensagem, para permitir que o Amazon SQS use o valor `DelaySeconds` do temporizador de mensagem em vez do valor `DelaySeconds` da fila.

Filas temporárias do Amazon SQS

Filas temporárias ajudam você a economizar tempo de desenvolvimento e custos de implantação ao usar padrões comuns de mensagens, como solicitação-resposta. Você pode usar o Temporary Queue Client para criar filas temporárias de alta taxa de transferência, econômicas e gerenciadas por aplicações.

O cliente mapeia várias filas temporárias (filas gerenciadas pela aplicação criadas sob demanda para um processo específico) em uma única fila do Amazon SQS automaticamente. Isso permite que o aplicativo faça menos chamadas de API e tenha uma taxa de transferência mais alta quando o tráfego é baixo para cada fila temporária. Quando uma fila temporária não está mais em uso, o cliente a limpa automaticamente, mesmo que alguns processos que usam o cliente não estejam encerrados corretamente.

Veja a seguir os benefícios das filas temporárias:

- Elas funcionam como canais de comunicação leves para segmentos ou processos específicos.
- É possível criá-las e excluí-las sem gerar custos adicionais.
- Elas são compatíveis em termos de API com filas estáticas (normais) do Amazon SQS. Isso significa que o código existente que envia e recebe mensagens pode enviar e receber mensagens de filas virtuais.

Tópicos

Filas virtuais

Filas virtuais são estruturas de dados locais criadas pelo Temporary Queue Client. As filas virtuais permitem combinar vários destinos de baixo tráfego em uma única fila do Amazon SQS. Para ver as práticas recomendadas, consulte Evitar reutilizar o mesmo ID de grupo de mensagens com filas virtuais.

> **Note**
>
> - Ao criar uma fila virtual, você cria apenas estruturas de dados temporárias nas quais os consumidores receberão as mensagens. Como não fazem chamadas de API para o Amazon SQS, as filas virtuais não geram custo.
> - As cotas de TPS se aplicam a todas as filas virtuais em uma única fila de host. Para ter mais informações, consulte Cotas de mensagens do Amazon SQS.

A classe de wrapper `AmazonSQSVirtualQueuesClient` adiciona suporte para atributos relacionados a filas virtuais. Para criar uma fila virtual, você deve chamar a ação de API `CreateQueue` usando o atributo `HostQueueURL`. Esse atributo especifica a fila existente que hospeda as filas virtuais.

O URL de uma fila virtual tem o formato a seguir.

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

Quando um produtor chama a ação de API `SendMessage` ou `SendMessageBatch` em um URL de fila virtual, o Temporary Queue Client faz o seguinte:

1. Extrai o nome da fila virtual.
2. Anexa o nome da fila virtual como um atributo de mensagem adicional.
3. Envia a mensagem para a fila de host.

Enquanto o produtor envia mensagens, um thread em segundo plano faz uma sondagem da fila de host e envia as mensagens recebidas para filas virtuais de acordo com os atributos de mensagem correspondentes.

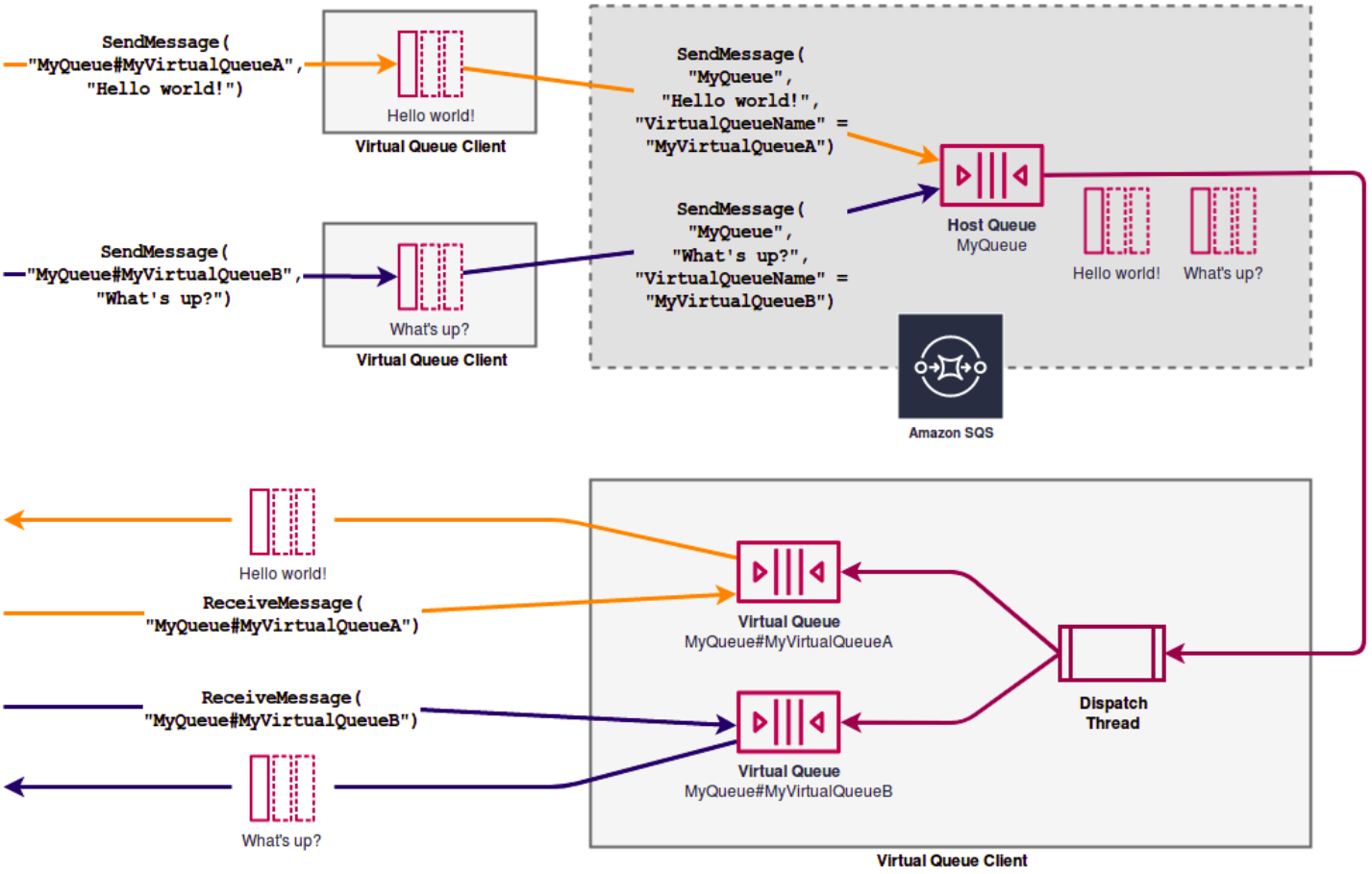
Enquanto o consumidor chama a ação de API `ReceiveMessage` em um URL de fila virtual, o Temporary Queue Client bloqueia a chamada localmente até o thread de fundo enviar uma mensagem para a fila virtual. (Esse processo é semelhante à pré-busca de mensagens no Cliente assíncrono armazenado em buffer: uma única ação de API pode fornecer mensagens para até 10 filas virtuais.) A exclusão de uma fila virtual remove todos os recursos do lado do cliente sem chamar o Amazon SQS.

A classe `AmazonSQSTemporaryQueuesClient` transforma automaticamente todas as filas criadas em filas temporárias. Ela também cria filas de host com os mesmos atributos automaticamente, sob demanda. Os nomes dessas filas compartilham o mesmo prefixo configurável (por padrão, `__RequesterClientQueues__`) que as identifica como filas temporárias. Isso permite que o cliente atue como uma substituição inicial que otimiza o código existente que cria e exclui filas. O cliente também inclui as interfaces `AmazonSQSRequester` e `AmazonSQSResponder` que permitem a comunicação bidirecional entre as filas.

Padrão de mensagens de resposta a solicitação (filas virtuais)

O caso de uso mais comum de filas temporárias é o padrão de mensagens de solicitação-resposta no qual um solicitante cria uma fila temporária para receber cada mensagem de resposta. Para evitar a criação de uma fila do Amazon SQS para cada mensagem de resposta, o Temporary Queue Client (cliente de fila temporária) permite criar e excluir várias filas temporárias sem fazer chamadas de API ao Amazon SQS. Para ter mais informações, consulte Implementação de sistemas de resposta a solicitação.

O diagrama a seguir mostra uma configuração comum usando esse padrão.



Cenário de exemplo: processar uma solicitação de login

O cenário de exemplo a seguir mostra como você pode usar as interfaces `AmazonSQSRequester` e `AmazonSQSResponder` para processar uma solicitação de login do usuário.

No lado do 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();
    }
}
```

O envio de uma solicitação de login faz o seguinte:

1. Cria uma fila temporária.
2. Anexa o URL da fila temporária à mensagem como um atributo.
3. Envia a mensagem.
4. Recebe uma resposta da fila temporária.
5. Exclui a fila temporária.
6. Retorna a resposta.

No lado do servidor

O exemplo a seguir pressupõe que, após a construção, um thread é criado para sondar a fila e chamar o método `handleLoginRequest()` para cada mensagem. Além disso, `doLogin()` é um método presumido.

```
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));
    }
}
```

Limpeza das filas

Para garantir que o Amazon SQS recupere todos os recursos na memória usados por filas virtuais, quando a aplicação não precisar mais do Temporary Queue Client, ele deverá chamar o método `shutdown()`. Você também pode usar o método `shutdown()` da interface `AmazonSQSRequester`.

O Temporary Queue Client também fornece uma maneira de eliminar filas de host órfãs. Para cada fila que recebe uma chamada de API durante um período (por padrão, cinco minutos), o cliente usa a ação de API `TagQueue` para etiquetar uma fila que permanece em uso.

> **Note**
>
> Qualquer ação de API executada em uma fila será marcada como não ociosa, inclusive uma ação `ReceiveMessage` que não retorne mensagens.

O thread em segundo plano usa as ações de API `ListQueues` e `ListTags` para verificar todas as filas com o prefixo configurado, excluindo as filas que não foram marcadas por pelo menos cinco minutos. Dessa forma, se um cliente não for encerrado corretamente, os outros clientes ativos serão limpos depois dele. A fim de reduzir a duplicação de trabalho, todos os clientes com o mesmo prefixo

se comunicam por meio de uma fila de trabalho interna compartilhada, denominada de acordo com o prefixo.

Temporizadores de mensagens do Amazon SQS

Os temporizadores de mensagem permitem que você especifique um período de invisibilidade inicial para uma mensagem adicionada a uma fila. Por exemplo, se você enviar uma mensagem com um temporizador de 45 segundos, a mensagem ficará invisível para os consumidores nos seus primeiros 45 segundos na fila. O atraso padrão (mínimo) para uma mensagem é 0 segundo. O máximo é 15 minutos. Para obter informações sobre o envio de mensagens com temporizadores usando o console, consulte Enviar uma mensagem.

> **Note**
>
> As filas FIFO não são compatíveis com temporizadores em mensagens individuais.

Para definir um período de atraso em uma fila inteira, em vez de mensagens individuais, use filas de atraso. Uma definição de temporizador de mensagem para uma mensagem individual substitui qualquer valor de `DelaySeconds` em uma fila de atraso do Amazon SQS.

Acessando o Amazon EventBridge Pipes por meio do console do Amazon SQS

O Amazon EventBridge Pipes conecta fontes a alvos. Os tubos são destinados a point-to-point integrações entre fontes e alvos suportados, com suporte para transformações e enriquecimento avançados. EventBridge O Pipes fornece uma maneira altamente escalável de conectar sua fila do Amazon SQS AWS a serviços como Step Functions, Amazon SQS e API Gateway, bem como a aplicativos de software como serviço (SaaS) de terceiros, como o Salesforce.

Para configurar um pipe, a origem é escolhida, adiciona filtragem opcional, define o enriquecimento opcional e escolhe o destino para os dados do evento.

Na página de detalhes de uma fila do Amazon SQS, você pode ver os pipes que usam essa fila como origem. Nessa página, você também pode:

- Inicie o EventBridge console para ver os detalhes do tubo.
- Inicie o EventBridge console para criar um novo canal com a fila como origem.

Para obter mais informações sobre como configurar uma fila do Amazon SQS como fonte de canal, consulte Amazon SQS queue as a source no Amazon User Guide. EventBridge Para obter mais informações sobre EventBridge tubos em geral, consulte EventBridge Tubos.

Para acessar EventBridge canais para uma determinada fila do Amazon SQS

1. Abra a página Queues (Filas) do console do Amazon SQS.
2. Selecione uma fila.
3. Na página de detalhes da fila, escolha a guia EventBridge Pipes.

   A guia EventBridge Pipes inclui uma lista de todos os canais atualmente configurados para usar a fila selecionada como fonte, incluindo:

   - nome do pipe
   - status atual
   - destino do pipe
   - última modificação do pipe
4. Veja mais detalhes do pipe ou crie um, se desejar:

   - Para acessar mais detalhes sobre um pipe:

     Selecione o nome do pipe.

     Isso abre a página de detalhes do Pipe do EventBridge console.
   - Para criar um pipe:

     Escolha Conectar fila do Amazon SQS ao pipe.

     Isso inicia a página Create pipe do EventBridge console, com a fila do Amazon SQS especificada como a origem do pipe. Para obter mais informações, consulte EventBridgeCriação de um tubo no Guia EventBridge do usuário da Amazon.

     > **Important**
     >
     > Uma mensagem em uma fila do Amazon SQS é lida por um único pipe e excluída da fila após ser processada, correspondendo ou não ao filtro configurado para o pipe. Tenha cuidado ao configurar vários pipes usando a mesma fila como origem.

Gerenciando grandes mensagens do Amazon SQS com a Extended Client Library e o Amazon Simple Storage Service

Você pode usar a Amazon SQS Extended Client Library para Java e a Amazon SQS Extended Client Library para Python para enviar mensagens grandes. Isso é especialmente útil para consumir grandes cargas de mensagens, de 256 KB a 2 GB. Ambas as bibliotecas salvam a carga da mensagem em um bucket do Amazon Simple Storage Service e enviam a referência do objeto Amazon S3 armazenado para a fila do Amazon SQS.

> **Note**
>
> As bibliotecas de clientes estendidas do Amazon SQS são compatíveis com as filas padrão e FIFO.

Tópicos

Gerenciando grandes mensagens do Amazon SQS usando Java e Amazon S3

Você pode usar a Amazon SQS Extended Client Library para Java e o Amazon Simple Storage Service (Amazon S3) para gerenciar grandes mensagens do Amazon Simple Queue Service (Amazon SQS). Isso é especialmente útil para consumir grandes cargas de mensagens, de 256 KB a 2 GB. A biblioteca salva a carga da mensagem em um bucket do Amazon S3 e envia uma mensagem contendo uma referência do objeto Amazon S3 armazenado para uma fila do Amazon SQS.

Você pode usar a biblioteca cliente Java estendida para o Amazon SQS para fazer o seguinte:

- Especificar se as mensagens são sempre armazenadas no Amazon S3 ou apenas quando uma mensagem tiver mais de 256 KB.
- Enviar uma mensagem que faça referência a um único objeto de mensagem armazenado em um bucket do S3.
- Recupere o objeto de mensagem de um bucket do Amazon S3

- Exclua o objeto de mensagem de um bucket do Amazon S3

Pré-requisitos

O exemplo a seguir usa o AWS Java SDK. Para instalar e configurar o SDK, consulte Configurar o AWS SDK para Java no Guia AWS SDK for Java do desenvolvedor.

Antes de executar o código de exemplo, configure suas AWS credenciais. Para obter mais informações, consulte Configurar AWS credenciais e região para desenvolvimento no Guia do AWS SDK for Java desenvolvedor.

O SDK for Java e a biblioteca cliente Java estendida para o Amazon SQS exigem o J2SE Development Kit 8.0 ou posterior.

> **Note**
>
> Você pode usar a biblioteca cliente Java estendida para o Amazon SQS para gerenciar mensagens do Amazon SQS usando o Amazon S3 somente com o AWS SDK for Java. Você não pode fazer isso com o AWS CLI console do Amazon SQS, a API HTTP do Amazon SQS ou qualquer outro SDKs. AWS

AWS Exemplo de SDK for Java 2.x: uso do Amazon S3 para gerenciar grandes mensagens do Amazon SQS

O exemplo de AWS SDK for Java 2.x a seguir cria um bucket do Amazon S3 com um nome aleatório e adiciona uma regra de ciclo de vida para excluir objetos permanentemente após 14 dias. Ele também cria uma fila chamada `MyQueue` e envia para ela uma mensagem aleatória, que é armazenada em um bucket do S3 e é maior que 256 KB. Por fim, o código recupera a mensagem, retorna informações sobre ela e, em seguida, exclui a mensagem, a fila e o 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 Exemplo de SDK for Java 2.x: uso do Amazon S3 para gerenciar grandes mensagens do Amazon SQS

O exemplo de AWS SDK for Java 2.x a seguir cria um bucket do Amazon S3 com um nome aleatório e adiciona uma regra de ciclo de vida para excluir objetos permanentemente após 14 dias. Ele também cria uma fila chamada `MyQueue` e envia para ela uma mensagem aleatória, que é armazenada em um bucket do S3 e é maior que 256 KB. Por fim, o código recupera a mensagem, retorna informações sobre ela e, em seguida, exclui a mensagem, a fila e o 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());
    }
}
```

Você pode usar o Apache Maven para configurar e criar o Amazon SQS Extended Client para seu projeto Java ou para criar o próprio SDK. Especifique módulos individuais do SDK que você usa em seu aplicativo.

```
<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>
```

Gerenciando grandes mensagens do Amazon SQS usando Python e Amazon S3

Você pode usar a biblioteca de cliente estendida do Amazon Simple Queue Service para Python e o Amazon Simple Storage Service para gerenciar grandes mensagens do Amazon SQS. Isso é especialmente útil para consumir grandes cargas de mensagens, de 256 KB a 2 GB. A biblioteca salva a carga da mensagem em um bucket do Amazon S3 e envia uma mensagem contendo uma referência do objeto Amazon S3 armazenado para uma fila do Amazon Amazon SQS.

Você pode usar a Extended Client Library for Python para fazer o seguinte:

- Especifique se as cargas são sempre armazenadas no Amazon S3 ou somente armazenadas no S3 quando o tamanho da carga excede 256 KB
- Envie uma mensagem que faça referência a um único objeto de mensagem armazenado em um bucket do Amazon S3
- Recupere o objeto de carga útil correspondente de um bucket do Amazon S3
- Exclua o objeto de carga útil correspondente de um bucket do Amazon S3

Pré-requisitos

A seguir estão os pré-requisitos para usar a Amazon SQS Extended Client Library para Python:

- Uma AWS conta com as credenciais necessárias. Para criar uma AWS conta, navegue até a página AWS inicial e escolha Criar uma AWS conta. Siga as instruções. Para obter informações sobre credenciais, consulte Credenciais.
- Um AWS SDK: o exemplo nesta página usa o AWS Python SDK Boto3. Para instalar e configurar o SDK, consulte a documentação do AWS SDK para Python no Guia do desenvolvedor do SDK AWS para Python

- Python 3.x (ou posterior) e. `pip`
- A biblioteca de cliente estendida do Amazon SQS para Python, disponível no PyPI

> **Note**
>
> Você pode usar a Amazon SQS Extended Client Library para Python para gerenciar mensagens do Amazon SQS usando o Amazon S3 somente com o SDK para Python. AWS Você não pode fazer isso com a AWS CLI, o console do Amazon SQS, a API HTTP do Amazon SQS ou qualquer outro SDKs. AWS

Configurar o armazenamento de mensagens

O Amazon SQS Extended Client usa os seguintes atributos de mensagem para configurar as opções de armazenamento de mensagens do Amazon S3:

- `large_payload_support`: O nome do bucket do Amazon S3 para armazenar mensagens grandes.
- `always_through_s3`: Se`True`, todas as mensagens serão armazenadas no Amazon S3. Nesse caso`False`, mensagens menores que 256 KB não serão serializadas no bucket s3. O padrão é `False`.
- `use_legacy_attribute`: Se`True`, todas as mensagens publicadas usarem o atributo de mensagem reservada antiga (`SQSLargePayloadSize`) em vez do atributo de mensagem reservada atual (`ExtendedPayloadSize`).

Gerenciando grandes mensagens do Amazon SQS com a Extended Client Library for Python

O exemplo a seguir cria um bucket do Amazon S3 com um nome aleatório. Em seguida, ele cria uma fila do Amazon SQS chamada `MyQueue` e envia uma mensagem que é armazenada em um bucket do S3 e tem mais de 256 KB para a fila. Por fim, o código recupera a mensagem, retorna informações sobre ela e, em seguida, exclui a mensagem, a fila e o 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
```

Configurando filas do Amazon SQS usando o console do Amazon SQS

Use o console do Amazon SQS para configurar e gerenciar filas e recursos do Amazon Simple Queue Service (Amazon SQS). Você também pode usar o console para configurar recursos como criptografia do lado do servidor, associar uma fila de mensagens mortas à sua fila ou definir um gatilho para invocar uma função. AWS Lambda

Tópicos

Controle de acesso baseado em atributos para Amazon SQS

O que é ABAC?

O controle de acesso baseado em atributos (ABAC) é um processo de autorização que define permissões com base em tags anexadas a usuários e recursos. AWS O ABAC fornece controle de acesso detalhado e flexível com base em atributos e valores, reduz o risco de segurança relacionado a políticas reconfiguradas baseadas em perfis e centraliza a auditoria e o gerenciamento de políticas de acesso. Para obter mais detalhes sobre o ABAC, consulte O que é a ABAC para a AWS, no Guia do usuário do IAM.

O Amazon SQS é compatível com o ABAC, permitindo que você controle o acesso às filas do Amazon SQS com base em tags e aliases associados a uma fila do Amazon SQS. As chaves de condição de tag e alias que ativam o ABAC no Amazon SQS autorizam as entidades principais a usar as filas do Amazon SQS sem editar políticas ou gerenciar concessões.

Com o ABAC, é possível usar tags para configurar permissões e políticas de acesso do IAM para as filas do Amazon SQS, o que ajuda você a escalar o gerenciamento de permissões. É possível criar uma única política de permissões no IAM usando tags que você adiciona a cada perfil empresarial, sem precisar atualizar a política toda vez em que adicionar um novo recurso. Você também pode anexar tags às entidades principais do IAM para criar uma política de ABAC. É possível criar políticas de ABAC para permitir operações do Amazon SQS quando a tag no perfil do usuário do IAM que está fazendo a chamada corresponde à tag de fila do Amazon SQS. Para saber mais sobre a marcação AWS, consulte Estratégias de AWS marcação e. Tags de alocação de custos do Amazon SQS

> **Note**
>
> Atualmente, o ABAC para Amazon SQS está disponível em AWS todas as regiões comerciais em que o Amazon SQS está disponível, com as seguintes exceções:
>
> - Ásia-Pacífico (Hyderabad)
> - Ásia-Pacífico (Melbourne)
> - Europa (Espanha)
> - Europa (Zurique)

Por que devo usar o ABAC no Amazon SQS?

Veja alguns benefícios de usar o ABAC no Amazon SQS:

- O ABAC para o Amazon SQS exige menos políticas de permissões. Não é necessário criar políticas diferentes para funções de trabalho diferentes. É possível usar tags de recurso e solicitação que se aplicam a mais de uma fila, o que reduz as despesas operacionais indiretas.
- Use o ABAC para escalar equipes rapidamente. As permissões para novos recursos são concedidas automaticamente com base em tags quando os recursos são devidamente marcados durante a criação.

- Use as permissões na entidade principal do IAM para restringir o acesso aos recursos. É possível criar tags para a entidade principal do IAM principal e usá-las para restringir o acesso a ações específicas que correspondam às tags na entidade principal do IAM. Isso ajuda você a automatizar o processo de concessão de permissões de solicitação.
- Acompanhe quem está acessando seus recursos. Você pode determinar a identidade de uma sessão examinando os atributos do usuário no AWS CloudTrail.

Tópicos

Chaves de condição do ABAC para o Amazon SQS

É possível usar as seguintes chaves de condição para controlar ações de funções:

Chave de condição para ABAC	Descrição	Tipo de política	Operações do Amazon SQS
leis: ResourceTag	A tag (chave e valor) na fila do Amazon SQS corresponde à tag (chave e valor) ou ao padrão de tag na política	Somente política do IAM	Operações de recursos de fila do Amazon SQS
leis: RequestTag	A tag (chave e valor) nas operações de recursos de fila do Amazon SQS corresponde à tag (chave e valor) ou ao padrão de tag na política	Política de fila e políticas do IAM	TagQueue, UntagQueue, CreateQueue

Chave de condição para ABAC	Descrição	Tipo de política	Operações do Amazon SQS
leis: TagKeys	Chaves de etiqueta na solicitação correspondem a chaves de etiqueta na política	Política de fila e políticas do IAM	TagQueue, UntagQueue, CreateQueue

Marcação para controle de acesso no Amazon SQS

Veja a seguir um exemplo de como usar tags para controle de acesso. A política do IAM restringe um usuário do IAM a todas as ações do Amazon SQS para todas as filas que incluem uma tag de recurso com a chave “environment” (ambiente) e o valor “production” (produção). Para obter mais informações, consulte Controle de acesso baseado em atributos com tags e Organizations AWS.

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

Criação de usuários do IAM e filas do Amazon SQS

Os exemplos a seguir explicam como criar uma política ABAC para controlar o acesso ao Amazon SQS usando e. AWS Management Console AWS CloudFormation

Usando o AWS Management Console

Criar um usuário do IAM

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. Selecione User (Usuário) no painel de navegação à esquerda.
3. Selecione Add Users (Adicionar usuários) e insira um nome na caixa de texto User name (Nome do usuário).
4. Escolha a caixa Access key - Programmatic access (Chave de acesso – Acesso programático) e selecione Next: Permissions (Próximo: permissões).
5. Selecione Next: Tags (Próximo: tags).
6. Adicione `beta` como a chave da tag e `environment` como o valor da tag.
7. Selecione Next: Review (Próximo: revisão) e, depois, Create user (Criar usuário).
8. Copie e armazene o ID da chave de acesso e a chave de acesso secreta em um local seguro.

Adicionar permissões de usuário do IAM

1. Selecione o usuário do IAM que você criou.
2. Escolha Add inline policy (Adicionar política em linha).
3. Na guia JSON, cole a seguinte política:

```
{
  "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. Escolha Revisar política.
5. Escolha Criar política.

Usando AWS CloudFormation

Use o AWS CloudFormation modelo de amostra a seguir para criar um usuário do IAM com uma política embutida anexada e uma fila do 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"
```

Testar o controle de acesso por atributo

Os exemplos a seguir mostram como testar o controle de acesso por atributo no Amazon SQS.

Crie uma fila com a chave da tag definida como “ambiente” e o valor da tag definido como “prod”

Execute esse comando da AWS CLI para testar a criação da fila com a chave de tag definida como environment e o valor da tag definido como prod. Se você não tiver a AWS CLI, poderá baixá-la e configurá-la para sua máquina.

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

Você recebe um erro `AccessDenied` do endpoint do Amazon SQS:

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

Isso ocorre porque o valor da tag no usuário do IAM não corresponde à tag transmitida na chamada da API `CreateQueue`. Lembre-se de que aplicamos uma tag ao usuário do IAM com a chave definida como `environment` e o valor definido como `beta`.

Crie uma fila com a chave da tag definida como “environment” e o valor da tag definido como “beta”

Execute esse comando da CLI para testar a criação de uma fila com a chave da tag definida como `environment` e o valor da tag definido como `beta`.

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

Você recebe uma mensagem confirmando a criação bem-sucedida da fila, semelhante à indicada abaixo.

```
{
"QueueUrl": "<queueUrl>“
}
```

Enviar uma mensagem para uma fila

Execute esse comando da CLI para testar o envio de uma mensagem a uma fila.

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

A resposta mostra uma entrega bem-sucedida de mensagens para a fila do Amazon SQS. A permissão de usuário do IAM permite que você envie uma mensagem para uma fila que tenha uma tag `beta`. A resposta inclui `MD5OfMessageBody` e `MessageId` contendo a mensagem.

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

Configurando parâmetros de fila usando o console do Amazon SQS

Quando você cria ou edita uma fila, pode configurar os seguintes parâmetros:

- Visibility timeout (Tempo limite de visibilidade): o período de tempo em que uma mensagem recebida de uma fila (por um consumidor) não estará visível para os outros consumidores de mensagens. Para obter mais informações, consulte Tempo limite de visibilidade.

  > **Note**
  >
  > Usar o console para definir o tempo limite de visibilidade configura o valor de tempo limite para todas as mensagens na fila. Para configurar o tempo limite para uma ou várias mensagens, você deve usar um dos AWS SDKs.

- Message retention period (Período de retenção de mensagens): a quantidade de tempo que o Amazon SQS retém as mensagens que permanecem na fila. Por padrão, uma fila retém mensagens por quatro dias. Você pode configurar uma fila para reter as mensagens por até 14 dias. Para obter mais informações, consulte Período de retenção de mensagens.
- Delivery delay (Atraso de entrega): quanto tempo o Amazon SQS atrasará antes de enviar uma mensagem adicionada à fila. Para obter mais informações, consulte Atraso de entrega.
- Maximum message size (Tamanho máximo da mensagem): tamanho máximo das mensagens para essa fila. Para obter mais informações, consulte Tamanho máximo da mensagem.
- Receive message wait time (Tempo de espera da mensagem): a quantidade máxima de tempo que o Amazon SQS espera para que as mensagens fiquem disponíveis depois que a fila recebe uma solicitação de recebimento. Para ter mais informações, consulte Sondagem curta e longa do Amazon SQS.
- Enable content-based deduplication (Habilitar a eliminação de duplicação baseada em conteúdo): o Amazon SQS pode criar automaticamente IDs de eliminação de duplicação com base no corpo da mensagem. Para ter mais informações, consulte Introdução às filas FIFO no Amazon SQS.
- Enable high throughput FIFO (Habilitar FIFO de alta taxa de transferência): use para habilitar a alta taxa de transferência disponível para mensagens na fila. Escolher esta opção altera as opções relacionadas (Deduplication scope [Escopo de eliminação de duplicação] e FIFO throughput limit [Limite de transferência FIFO]) para as configurações necessárias a fim de habilitar a alta taxa de transferência para filas FIFO. Para obter mais informações, consulte Alta taxa de transferência para filas FIFO no Amazon SQS e Cotas de mensagens do Amazon SQS.

- Redrive allow policy (Política de permissão de redirecionamento): define quais filas de origem podem usar essa fila como a fila de mensagens mortas. Para ter mais informações, consulte Usando filas de mensagens mortas no Amazon SQS .

Para configurar os parâmetros de uma fila existente (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues. Escolha uma fila e escolha Edit (Editar).
3. Role até a seção Configuration (Configuração).
4. Em Visibility timeout (Tempo limite de visibilidade), insira a duração e as unidades. O intervalo é de 0 segundo a 12 horas. O valor padrão de é 30 segundos.
5. Em Message retention period (Período de retenção de mensagens), insira a duração e as unidades. O intervalo é de 1 minuto a 14 dias. O valor padrão é 4 dias.
6. Em uma fila padrão, insira um valor para Receive message wait time (Tempo de espera da mensagem). O intervalo é de 0 a 20 segundos. O valor padrão é 0 segundo, o que define uma sondagem curta. Qualquer valor diferente de zero define uma sondagem longa.
7. Em Delivery delay (Atraso de entrega), insira a duração e as unidades. O intervalo é de 0 segundo a 15 minutos. O valor de padrão é 0 segundos.
8. Em Maximum message size (Tamanho máximo da mensagem), insira um valor. O intervalo é de 1 KB a 256 KB. O valor padrão é 256 KB.
9. Para uma fila FIFO, escolha Enable content-based deduplication (Habilitar a eliminação de duplicação baseada em conteúdo) para habilitar a eliminação de duplicação baseada em conteúdo. Por padrão, essa configuração está desabilitada.
10. (Opcional) Em uma fila FIFO, para habilitar um throughput mais alto a fim de enviar e receber mensagens na fila, escolha Enable high throughput FIFO (Habilitar FIFO de alto throughput).

    Escolher esta opção altera as opções relacionadas (Deduplication scope [Escopo de eliminação de duplicação] e FIFO throughput limit [Limite de transferência FIFO]) para as configurações necessárias a fim de habilitar a alta taxa de transferência para filas FIFO. Se você alterar qualquer uma das configurações necessárias para usar FIFO de alta taxa de transferência, a taxa de transferência normal permanecerá em vigor para a fila e a eliminação de duplicação ocorrerá conforme especificado. Para obter mais informações, consulte Alta taxa de transferência para filas FIFO no Amazon SQS e Cotas de mensagens do Amazon SQS.

11. Em Redrive allow policy (Política de permissão de redirecionamento), escolha Enabled (Habilitada). Selecione uma das seguintes opções: Allow all (Permitir tudo) (padrão), By queue (Por fila) ou Deny all (Negar tudo). Ao escolher By queue (Por fila), especifique uma lista de até 10 filas de origem pelo nome do recurso da Amazon (ARN).
12. Quando você terminar de configurar os parâmetros da fila, escolha Save (Salvar).

Configurar políticas de acesso

Quando você edita uma fila, você pode configurar sua política de acesso.

A política de acesso define as contas, usuários e funções que podem acessar a fila. A política de acesso também define as ações (como `SendMessage`, `ReceiveMessage` ou `DeleteMessage`) que os usuários podem acessar. A política padrão permite que apenas o proprietário da fila envie e receba mensagens.

Para configurar a política de acesso para uma fila existente (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Escolha uma fila e escolha Edit (Editar).
4. Role até a seção Access policy (Política de acesso).
5. Edite as instruções de política de acesso na caixa de entrada. Para obter mais informações sobre instruções da política de acesso, consulteGerenciamento de identidade e acesso no Amazon SQS.
6. Quando terminar de configurar a política de acesso, escolha Save (Salvar).

Configurando a criptografia do lado do servidor para uma fila usando chaves de criptografia gerenciadas pelo SQS

Além da opção padrão da criptografia do lado do servidor (SSE) gerenciada pelo Amazon SQS, a SSE gerenciada pelo Amazon SQS (SSE-SQS) permite criar uma criptografia do lado do servidor gerenciada pelo cliente que usa chaves de criptografia gerenciadas pelo SQS para proteger dados sigilosos enviados por filas de mensagens. Com o SSE-SQS, você não precisa criar e gerenciar chaves de criptografia ou modificar seu código para criptografar seus dados. O SSE-SQS permite

transmitir dados com segurança e ajuda a atender a requisitos regulamentares e conformidade com criptografia rigorosa sem custo adicional.

O SSE-SQS protege os dados em repouso usando a criptografia Advanced Encryption Standard (AES-256) de 256 bits. A SSE criptografa mensagens assim que o Amazon SQS as recebe. O Amazon SQS armazena as mensagens no formato criptografado e as descriptografa apenas quando elas são enviadas a um consumidor autorizado.

> **Note**
>
> - A opção de SSE comum só é efetiva quando você cria uma fila sem especificar atributos de criptografia.
> - O Amazon SQS permite que você desative toda a criptografia de filas. Portanto, desativar a KMS-SSE não habilitará automaticamente a SQS-SSE. Se quiser habilitar a SQS-SSE depois de desativar a KMS-SSE, você deverá adicionar uma alteração de atributos na solicitação.

Para configurar a criptografia SSE-SQS para uma fila (console)

> **Note**
>
> Qualquer fila criada usando o endpoint HTTP (não TLS) não habilitará a criptografia SSE-SQS por padrão. É uma prática recomendada de segurança criar filas do Amazon SQS usando endpoints HTTPS ou do Signature versão 4.

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Escolha uma fila e escolha Edit (Editar).
4. Expanda Encryption (Criptografia).
5. Em Server-side encryption (Criptografia do lado do servidor), escolha Enabled (Habilitado) (padrão).

> **Note**
>
> Com a SSE habilitada, as solicitações anônimas `SendMessage` e `ReceiveMessage` à fila criptografada serão rejeitadas. As práticas recomendadas de segurança do Amazon SQS não aconselham o uso de solicitações anônimas. Se você quiser enviar solicitações anônimas a uma fila do Amazon SQS, desabilite o SSE.

6. Selecione Amazon SQS key (SSE-SQS) (Chave do Amazon SQS (SSE-SQS). Não há custo adicional para usar essa opção.
7. Escolha Salvar.

Configurando a criptografia do lado do servidor para uma fila usando o console do Amazon SQS

Para proteger os dados nas mensagens de uma fila, o Amazon SQS tem a criptografia do lado do servidor (SSE) habilitada por padrão para todas as filas recém-criadas. O Amazon SQS integra-se ao Amazon Web Services Key Management Service (Amazon Web Services KMS) para gerenciar chaves do KMS para criptografia do lado do servidor (SSE). Para obter mais informações sobre o uso de SSE, consulte Criptografia em repouso no Amazon SQS.

A chave do KMS que você atribui à fila deve ter uma política de chave que inclua permissões para todas as entidades autorizadas a usar a fila. Para obter mais informações, consulte Gerenciamento de chaves.

Se você não for o proprietário da chave do KMS ou se fizer login com uma conta que não tenha as permissões `kms:ListAliases` e `kms:DescribeKey`, não será possível visualizar as informações sobre a chave do KMS no console do Amazon SQS. Peça ao proprietário da chave do KMS para conceder essas permissões a você. Para obter mais informações, consulte Gerenciamento de chaves.

Quando você cria ou edita uma fila, pode configurar a SSE-KMS.

Para configurar a SSE-KMS para uma fila existente (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.

3. Escolha uma fila e escolha Edit (Editar).
4. Expanda Encryption (Criptografia).
5. Em Server-side encryption (Criptografia do lado do servidor), escolha Enabled (Habilitado) (padrão).

> **Note**
>
> Com a SSE habilitada, as solicitações anônimas `SendMessage` e `ReceiveMessage` à fila criptografada serão rejeitadas. As práticas recomendadas de segurança do Amazon SQS não aconselham o uso de solicitações anônimas. Se você quiser enviar solicitações anônimas a uma fila do Amazon SQS, desabilite o SSE.

6. Selecione AWS Key Management Service key (SSE-KMS) (Chave do serviço de gerenciamento de chaves (SSE-KMS).

   O console exibe a Descrição, a Conta e o ARN da chave do KMS da chave do KMS.
7. Especifique o ID da chave do KMS para a fila. Para ter mais informações, consulte Principais termos.

   a. Escolha a opção Choose a KMS key alias (Escolha um alias para a chave do KMS).

   b. A chave padrão é a chave do KMS gerenciada pela Amazon Web Services para o Amazon SQS. Para usar essa chave, escolha-a na lista KMS key (Chave do KMS).

   c. Para usar uma chave do KMS personalizada de sua conta do Amazon Web Services, escolha-a na lista KMS key (Chave do KMS). Para obter instruções sobre como criar chaves do KMS personalizadas, consulte Creating Keys (Criar chaves) no Amazon Web Services Key Management Service Developer Guide (Guia do desenvolvedor do serviço de gerenciamento de chaves do Amazon Web Services).

   d. Para usar uma chave do KMS personalizada que não esteja na lista ou uma chave do KMS personalizada de outra conta do Amazon Web Services, escolha Enter the KMS key alias (Inserir o alias da chave do KMS) e insira o nome do recurso da Amazon (ARN) da chave do KMS.
8. (Opcional) Em Data key reuse period (Período de reutilização de chaves de dados), especifique um valor entre 1 minuto e 24 horas. O padrão é 5 minutos. Para ter mais informações, consulte Entender o período de reutilização de chaves de dados.
9. Quando terminar de configurar a SSE-KMS, escolha Save (Salvar).

Configurando tags de alocação de custos para uma fila usando o console do Amazon SQS

Você pode adicionar tags de alocação de custos às suas filas do Amazon SQS para ajudar a organizá-las e identificá-las. Para ter mais informações, consulte Tags de alocação de custos do Amazon SQS.

Na página Details (Detalhes) de uma fila, a guia Tagging (Marcação) exibe as etiquetas para a fila.

Quando você cria ou edita uma fila, pode configurar tags para ela.

Para configurar tags para uma fila existente (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Escolha uma fila e escolha Edit (Editar).
4. Role até a seção Tags.
5. Adicionar, modificar ou remover tags da fila:
    a. Para adicionar uma tag, escolha Add new tag (Adicionar nova tag), insira uma Key (Chave) e um Value (Valor) e, em seguida, escolha Add new tag (Adicionar nova tag).
    b. Para atualizar uma tag, altere sua Key (Chave) e Value (Valor).
    c. Para remover uma tag, escolha Remove tag (Remover tag) ao lado de um par chave-valor.
6. Ao terminar de configurar as tags, escolha Save (Salvar).

Inscrever uma fila em um tópico do Amazon SNS usando o console do Amazon SQS

Você pode assinar uma ou mais filas do Amazon SQS em um tópico do Amazon Simple Notification Service (Amazon SNS). Quando você publica uma mensagem em um tópico, o Amazon SNS envia a mensagem para cada fila inscrita. O Amazon SQS gerencia a assinatura e todas as permissões necessárias. Para ter mais informações sobre o Amazon SNS, consulte O que é o Amazon SNS? no Guia do desenvolvedor do Amazon Simple Notification Service.

Quando você assina uma fila do Amazon SQS, em um tópico do SNS, o Amazon SNS usa HTTPS para encaminhar mensagens para o Amazon SQS. Para obter informações sobre como usar o

Amazon SNS com filas criptografadas do Amazon SQS, consulte Configurar permissões KMS para serviços AWS.

> **Important**
>
> O Amazon SQS é compatível com até 20 instruções por política de acesso. Assinar um tópico do Amazon SNS adiciona uma instrução desse tipo. Exceder esse valor causará uma falha na entrega da assinatura do tópico.

Para inscrever uma fila em um tópico do SNS (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Na lista de filas, escolha a fila (ou filas) que você deseja assinar em um tópico do SNS.
4. Em Actions (Ações), escolha Subscribe to Amazon SNS topic (Inscrever-se no tópico do Amazon SNS).
5. No menu Specify an Amazon SNS topic available for this queue (Especificar um tópico do Amazon SNS disponível para esta fila), escolha o tópico do SNS para sua fila.

   Se o tópico do SNS não estiver listado no menu, escolha Enter Amazon SNS topic ARN (Inserir o ARN do tópico do Amazon SNS) e, em seguida, insira o nome do recurso da Amazon (ARN) do tópico.
6. Escolha Salvar.
7. Para verificar o resultado da assinatura, publique no tópico e, em seguida, visualize a mensagem que o tópico envia para a fila. Para ter mais informações, consulte Publicação de mensagens do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Se a fila do Amazon SQS e o tópico do SNS forem diferentes Contas da AWS, o proprietário do tópico deverá primeiro confirmar a assinatura. Para obter mais informações, consulte Confirme a assinatura no Guia do desenvolvedor do Amazon Simple Storage Service.

Para obter informações sobre a assinatura de um tópico de SNS entre regiões, consulte Envio de mensagens do Amazon SNS para uma fila ou função do Amazon SQS em uma região diferente AWS Lambda no Guia do desenvolvedor do Amazon Simple Notification Service

Configurando uma fila do Amazon SQS para acionar uma função AWS Lambda

Você pode usar uma AWS Lambda função para processar mensagens em uma fila do Amazon SQS. O Lambda sonda a fila e invoca sua função Lambda de forma síncrona com um evento que contém mensagens da fila. Para permitir que a função tenha tempo para processar cada lote de registros, defina o tempo limite de visibilidade da fila de origem para pelo menos seis vezes o tempo limite configurado na sua função. Esse tempo extra permite que o Lambda repita o processo se a execução da sua função for limitada durante o processamento de um lote anterior.

Você pode especificar outra fila para atuar como uma fila de mensagens mortas para mensagens que sua função Lambda não pode processar.

Uma função Lambda pode processar itens de várias filas (uma fonte de eventos do Lambda para cada fila). É possível usar a mesma fila com várias funções Lambda.

Se você associar uma fila criptografada a uma função Lambda, mas o Lambda não sondar as mensagens, adicione a permissão `kms:Decrypt` para sua função de execução do Lambda.

Observe as seguintes restrições:

- Sua fila e a função Lambda devem estar na mesma região AWS .
- Uma fila criptografada que usa a chave padrão (chave KMS AWS gerenciada para Amazon SQS) não pode invocar uma função Lambda em outra. Conta da AWS

Para obter informações sobre a implementação da função Lambda, consulte Como usar AWS Lambda com o Amazon SQS no AWS Lambda Guia do desenvolvedor.

Pré-requisitos

Para configurar os acionadores de função Lambda, você deve atender aos seguintes requisitos:

- Se você usar um usuário, o perfil do Amazon SQS deverá incluir as seguintes permissões:
  - `lambda:CreateEventSourceMapping`
  - `lambda:ListEventSourceMappings`
  - `lambda:ListFunctions`
- A função de execução do Lambda deve incluir as seguintes permissões:

- `sqs:DeleteMessage`
- `sqs:GetQueueAttributes`
- `sqs:ReceiveMessage`

- Se você associar uma fila criptografada a uma função Lambda, adicione a permissão `kms:Decrypt` à função de execução do Lambda.

Para ter mais informações, consulte Visão geral do gerenciamento de acesso no Amazon SQS.

Para configurar uma fila para acionar uma função Lambda (console)

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Na página Queues (Filas), escolha a fila a ser configurada.
4. Na página da fila, escolha a guia Lambda triggers (Acionadores do Lambda).
5. Na página Lambda triggers (Acionadores do Lambda), escolha um acionador do Lambda.

   Se a lista não incluir o acionador do Lambda de que você precisa, escolha Configure Lambda function trigger (Configurar acionador da função Lambda). Insira o nome do recurso da Amazon (ARN) da função Lambda ou escolha um recurso existente. Em seguida, escolha Salvar.
6. Escolha Salvar. O console salva a configuração e exibe a página Details (Detalhes) da fila.

   Na página Details (Detalhes), a guia Lambda triggers (Acionadores do Lambda) exibe a função Lambda e seu status. Demora aproximadamente um minuto para a função Lambda se associar à sua fila.
7. Para verificar os resultados da configuração, você pode enviar uma mensagem à fila e, em seguida, visualizar a função Lambda acionada no console do Lambda.

Automatização de notificações de AWS serviços para o Amazon SQS usando a Amazon EventBridge

A Amazon EventBridge permite automatizar AWS serviços e responder a eventos do sistema, como problemas de disponibilidade de aplicativos ou alterações de recursos. Os eventos dos AWS serviços são entregues EventBridge quase em tempo real. É possível criar regras simples para indicar quais eventos são de seu interesse, e quais ações automatizadas devem ser tomadas quando um evento corresponder a uma regra.

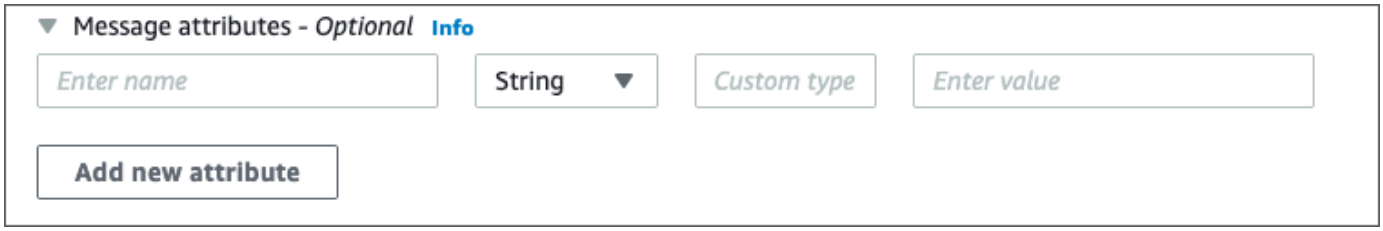
EventBridge permite que você defina uma variedade de destinos, como o padrão Amazon SQS e as filas FIFO, que recebem eventos no formato JSON. Para obter mais informações, consulte as EventBridgemetas da Amazon no Guia EventBridge do usuário da Amazon.

Enviar uma mensagem com atributos

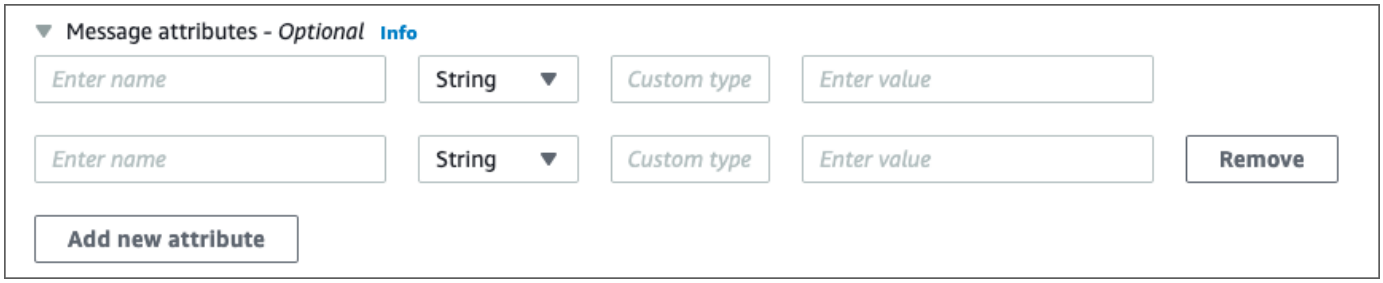
Para filas FIFO e padrão, é possível incluir metadados estruturados (como carimbos de data e hora, dados geoespaciais, assinaturas e identificadores) com mensagens. Para ter mais informações, consulte Atributos de mensagem do Amazon SQS.

Para enviar uma mensagem com atributos para uma fila usando o console do Amazon SQS

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. No painel de navegação, escolha Queues.
3. Na página Queues (Filas), escolha uma fila.
4. Escolha Send and receive messages (Enviar e receber mensagens).
5. Insira os parâmetros do atributo de mensagem.
   a. Na caixa de texto de nome, insira um nome exclusivo de até 256 caracteres.
   b. Para o tipo de atributo, escolha String, Number (Número) ou Binary (Binário).
   c. (Opcional) Insira um tipo de dado personalizado. Por exemplo, você pode adicionar **`byte`**, **`int`** ou **`float`** como tipos de dados personalizados para Number (Número).
   d. Na caixa de texto do valor, insira o valor do atributo de mensagem.



6. Para adicionar outro atributo de mensagem, escolha Add new attribute (Adicionar um atributo).



7. Você pode modificar os valores do atributo a qualquer momento antes de enviar a mensagem.
8. Para excluir um atributo, escolha Remove (Remover). Para excluir o primeiro atributo, feche Message attributes (Atributos de mensagem).
9. Ao concluir a adição de atributos à mensagem, escolha Send Message (Enviar mensagem). Sua mensagem é enviada e o console exibe uma mensagem de sucesso. Para visualizar informações sobre os atributos da mensagem enviada, escolha View details (Visualizar detalhes). Escolha Done (Concluído) para fechar a caixa de diálogo Message details (Detalhes da mensagem).

Práticas recomendadas do Amazon SQS

Estas práticas recomendadas podem ajudar você a aproveitar ao máximo o Amazon SQS.

Tópicos

Recomendações para filas FIFO e padrão do Amazon SQS

As práticas recomendadas a seguir podem ajudar a reduzir custos e a processar mensagens com eficiência usando o Amazon SQS.

Tópicos

Trabalhar com mensagens do Amazon SQS

As diretrizes a seguir podem ajudar a processar mensagens com eficiência usando o Amazon SQS.

Tópicos

Processar mensagens em tempo hábil

Definir o tempo limite de visibilidade depende do tempo de que o seu aplicativo precisa para processar e excluir uma mensagem. Por exemplo, se seu aplicativo exigir 10 segundos para processar uma mensagem e você definir o tempo limite de visibilidade como 15 minutos, deverá esperar por um tempo relativamente longo para tentar processar a mensagem novamente se a tentativa de processamento anterior falhar. Como alternativa, se o aplicativo exigir 10 segundos para processar uma mensagem, mas você definir o tempo limite de visibilidade como apenas 2 segundos, uma mensagem duplicada será recebida por outro consumidor enquanto o consumidor original ainda estiver trabalhando na mensagem.

Para garantir que haja tempo suficiente para processar mensagens, use uma das seguintes estratégias:

- Se você souber (ou puder razoavelmente estimar) quanto tempo leva para processar uma mensagem, amplie o tempo limite de visibilidade da mensagem para o máximo de tempo necessário para processar e excluir a mensagem. Para obter mais informações, consulte Configurar tempo limite de visibilidade.
- Se você não souber quanto tempo leva para processar uma mensagem, crie uma pulsação para o processo do consumidor: especifique o tempo limite de visibilidade inicial (por exemplo, 2 minutos) e, desde que o consumidor ainda funcione na mensagem, continue estendendo o tempo limite de visibilidade em 2 minutos a cada minuto.

> **Important**
>
> O tempo limite de visibilidade máximo é de 12 horas a partir do momento em que o Amazon SQS recebe `ReceiveMessage`. Estender o tempo limite de visibilidade não redefine o período máximo de 12 horas.
> Além disso, talvez você não consiga definir o tempo limite em uma mensagem individual para as 12 horas completas (por exemplo, 43.200 segundos), pois a solicitação de `ReceiveMessage` inicia o temporizador. Por exemplo, se você receber uma mensagem e definir imediatamente o máximo de 12 horas enviando uma chamada de `ChangeMessageVisibility` com `VisibilityTimeout` igual a 43.200 segundos, provavelmente ocorrerá uma falha. No entanto, usar um valor de 43.195 segundos funcionará, a menos que haja um atraso significativo entre a solicitação da mensagem via `ReceiveMessage` e a atualização do tempo limite de visibilidade. Se o consumidor precisar de mais de 12 horas, considere usar o Step Functions.

Lidar com erros de solicitação

Para gerenciar erros de solicitação, use uma das seguintes estratégias:

- Se você usa um AWS SDK, já tem uma lógica automática de repetição e recuo à sua disposição. Para ter mais informações, consulte Repetições de erro e recuo exponencial na AWS no Referência geral da Amazon Web Services.
- Se você não usa os recursos do AWS SDK para tentar novamente e recuar, faça uma pausa (por exemplo, 200 ms) antes de tentar novamente a ReceiveMessageação depois de não receber nenhuma mensagem, um tempo limite ou uma mensagem de erro do Amazon SQS. Para o uso subsequente de `ReceiveMessage` que oferece os mesmos resultados, faça uma pausa maior (por exemplo, 400 ms).

Configuração da sondagem longa

Quando o tempo de espera da ação da API `ReceiveMessage` é maior do que 0, a sondagem longa está em vigor. O tempo máximo de espera de sondagem longa é de 20 segundos. A sondagem longa ajuda a reduzir os custos de uso do Amazon SQS eliminando o número de respostas vazias (quando não há mensagens disponíveis para uma solicitação `ReceiveMessage`) e respostas vazias falsas (quando mensagens estão disponíveis, mas não são incluídas em uma resposta). Para ter mais informações, consulte Sondagem curta e longa do Amazon SQS.

Para garantir o processamento ideal de mensagens, use as seguintes estratégias:

- Na maioria dos casos, você pode definir o tempo de espera de `ReceiveMessage` como 20 segundos. Se 20 segundos for muito longo para seu aplicativo, defina um tempo de espera `ReceiveMessage` mais curto (no mínimo, 1 segundo). Se você não usa um AWS SDK para acessar o Amazon SQS, ou se configura AWS um SDK para ter um tempo de espera menor, talvez seja necessário modificar seu cliente Amazon SQS para permitir solicitações mais longas ou usar um tempo de espera menor para pesquisas longas.
- Se você implementar a sondagem longa para várias filas, use um thread para cada fila, em vez de um único thread para todas as filas. O uso de um único thread para cada fila permite que seu aplicativo processe as mensagens em cada uma das filas conforme se tornam disponíveis, enquanto o uso de um único thread para sondar várias filas pode fazer com que seu aplicativo não possa processar as mensagens disponíveis em outras filas enquanto o aplicativo aguarda (até 20 segundos) por uma fila que não tem mensagens disponíveis.

> **Important**
>
> Para evitar erros de HTTP, certifique-se de que o tempo limite da resposta HTTP para solicitações `ReceiveMessage` é maior do que o parâmetro `WaitTimeSeconds`. Para obter mais informações, consulte ReceiveMessage.

Capturar mensagens problemáticas

Para capturar todas as mensagens que não podem ser processadas e coletar CloudWatch métricas precisas, configure uma fila de mensagens mortas.

- A política de redirecionamento redireciona mensagens para uma dead letter queue depois que a fila de origem falha em processar uma mensagem um número de vezes especificado.
- O uso da dead letter queue diminui o número de mensagens e reduz a possibilidade de exposição a mensagens poison pill (mensagens que podem ser recebidas, mas que não podem ser processadas).
- Incluir uma mensagem de pílula venenosa em uma fila pode distorcer a `ApproximateAgeOfOldestMessage` CloudWatch métrica, fornecendo uma idade incorreta da mensagem da pílula venenosa. Configurar uma dead letter queue ajuda a evitar alarmes falsos ao usar essa métrica.

Configurando a retenção de filas de mensagens mortas

Para filas comuns, a validade de uma mensagem é sempre baseada em seu carimbo de data/hora de enfileiramento original. Quando uma mensagem é movida para uma fila de mensagens mortas, o carimbo de data/hora de enfileiramento permanece inalterado. A métrica `ApproximateAgeOfOldestMessage` indica quando a mensagem foi movida para a fila de mensagens mortas, não quando a mensagem foi originalmente enviada. Por exemplo, suponha que uma mensagem fique um dia na fila original antes de ser movida para uma fila de mensagens mortas. Se o período de retenção da fila de mensagens mortas for de quatro dias, a mensagem será excluída da fila de mensagens mortas após três dias e a `ApproximateAgeOfOldestMessage` será de três dias. Portanto, é uma prática recomendada definir sempre o período de retenção de uma fila de mensagens mortas para ser maior do que o período de retenção da fila original.

Para filas FIFO, o carimbo de data/hora da fila é redefinido quando a mensagem é movida para uma fila de mensagens não entregues. A métrica `ApproximateAgeOfOldestMessage` indica

quando a mensagem foi movida para a fila de mensagens não entregues. No mesmo exemplo acima, a mensagem é excluída da fila de mensagens não entregues após quatro dias e `ApproximateAgeOfOldestMessage` é de quatro dias.

Evitar o processamento inconsistente de mensagens

Como o Amazon SQS é um sistema distribuído, é possível que um consumidor não receba uma mensagem mesmo quando o Amazon SQS a marca como entregue ao retornar com êxito de uma chamada de método da API `ReceiveMessage`. Nesse caso, o Amazon SQS registra a mensagem como entregue pelo menos uma vez, embora o consumidor nunca a tenha recebido. Como nenhuma tentativa adicional de entregar mensagens é feita sob essas condições, não recomendamos definir o número máximo de recebimentos como 1 para uma fila de mensagens mortas.

Implementação de sistemas de resposta a solicitação

Ao implementar um sistema de solicitação-resposta ou chamada de procedimento remoto (RPC), lembre-se das seguintes melhores práticas:

- Não crie filas de respostas por mensagem. Em vez disso, crie filas de resposta na inicialização, por produtor, e use um atributo de mensagem de ID de correlação para mapear respostas às solicitações.
- Não permita que os produtores compartilhem filas de respostas. Isso pode fazer com que um produtor receba mensagens de resposta destinadas a outro produtor.

Para obter mais informações sobre a implementação do padrão de solicitação-resposta usando o Temporary Queue Client, consulte Padrão de mensagens de resposta a solicitação (filas virtuais).

Reduzir os custos do Amazon SQS

As melhores práticas a seguir podem ajudar a reduzir custos e a aproveitar outras possibilidades de redução de custos e obter resposta quase instantânea.

Agrupar ações de mensagem em lotes

Para reduzir custos, coloque suas ações de mensagem em lotes:

- Para enviar, receber e excluir mensagens, e para alterar o tempo limite de visibilidade de várias mensagens com uma única ação, use as ações da API em lotes do Amazon SQS.

- Para combinar o armazenamento em buffer no lado do cliente com o envio de solicitações em lotes, use a sondagem longa junto com l cliente assíncrono armazenado em buffer incluído com o AWS SDK for Java.

> **Note**
>
> Atualmente, o cliente assíncrono no buffer do Amazon SQS não oferece suporte a filas FIFO.

Usar o modo de sondagem apropriado

- A sondagem longa permite que você consuma mensagens da fila do Amazon SQS assim que elas se tornam disponíveis.
  - Para reduzir o custo do uso do Amazon SQS e diminuir o número de recebimentos vazios em uma fila vazia (respostas à ação `ReceiveMessage` que não retornam nenhuma mensagem), habilite a sondagem longa. Para obter mais informações, consulte Sondagem longa do Amazon SQS.
  - Para aumentar a eficiência ao sondar vários threads com vários recebimentos, diminua o número de threads.
  - A sondagem longa é melhor do que a sondagem curta na maioria dos casos.
- A sondagem curta retorna respostas imediatamente, mesmo que a fila do Amazon SQS sondada esteja vazia.
  - Para satisfazer os requisitos de um aplicativo que espera respostas imediatas para a solicitação `ReceiveMessage`, use a sondagem curta.
  - A sondagem curta é cobrada pelo mesmo custo de uma sondagem longa.

Migrar de uma fila padrão do Amazon SQS para uma fila FIFO

Se você não estiver configurando o parâmetro `DelaySeconds` de cada mensagem, poderá migrar para uma fila FIFO fornecendo um ID de grupo de mensagens para cada mensagem enviada.

Para ter mais informações, consulte Migração de uma fila padrão para uma fila FIFO no Amazon SQS.

Recomendações adicionais para filas FIFO do Amazon SQS

As práticas recomendadas a seguir podem ajudar você a usar o ID de eliminação de duplicação de mensagens e o ID de grupo de mensagens de forma ideal. Para obter mais informações, consulte as ações `SendMessage` e `SendMessageBatch` na Referência da API do Amazon Simple Queue Service.

Tópicos

Usar o ID de eliminação de duplicação de mensagens do Amazon SQS

O ID de eliminação de duplicação de mensagens é o token usado para a eliminação de duplicação de mensagens enviadas. Se uma mensagem com um ID de eliminação de duplicação de mensagens específico for enviada com êxito, todas as mensagens enviadas com o mesmo ID de eliminação de duplicação de mensagens serão aceitas com êxito, mas não serão entregues durante o intervalo de eliminação de duplicação de cinco minutos.

> **Note**
>
> O Amazon SQS continua acompanhando o ID de eliminação de duplicação da mensagem mesmo depois que a mensagem é recebida e excluída.

Fornecer o ID de eliminação de duplicação de mensagens

O produtor deve fornecer valores de ID de eliminação de duplicação de mensagens para cada mensagem nos seguintes cenários:

- Mensagens enviadas com corpos idênticos, mas que o Amazon SQS deve tratar como únicas.
- Mensagens enviadas com conteúdo idêntico, mas com diferentes atributos, que o Amazon SQS deve tratar como únicas.
- Mensagens enviadas com conteúdo diferente (por exemplo, contagens de repetições incluídas no corpo da mensagem), que o Amazon SQS deve tratar como duplicações.

Habilitar a eliminação de duplicação para um sistema de produtor/consumidor único

Se você tiver um único produtor e um único consumidor e as mensagens forem exclusivas porque um ID de mensagem específico do aplicativo foi incluído no corpo da mensagem, siga estas melhores práticas:

- Ative a eliminação de duplicação baseada em conteúdo para a fila (cada uma de suas mensagens tem um único corpo). O produtor pode omitir o ID de eliminação de duplicação de mensagem.
- Quando a desduplicação baseada em conteúdo é habilitada para uma fila FIFO do Amazon SQS e uma mensagem é enviada com uma ID de desduplicação, a ID de desduplicação substitui a ID de desduplicação baseada em conteúdo geradaSendMessage.
- Embora o consumidor não seja obrigado a fornecer um ID de tentativa de solicitação de recebimento, isso é uma prática recomendada porque permite que sequências de tentativa de recuperação de falhas sejam executadas mais rapidamente.
- Você pode tentar enviar ou receber solicitações novamente, porque elas não interferem na ordenação de mensagens em filas FIFO.

Projetar para cenários de recuperação de interrupção

O processo de eliminação de duplicação em filas FIFO é depende do tempo. Ao projetar sua aplicação, garanta que o produtor e o consumidor possam se recuperar em caso de interrupção de comunicação de um cliente ou da rede.

- O produtor deve estar ciente do intervalo de eliminação de duplicação da fila. O Amazon SQS tem um intervalo de eliminação de duplicação de cinco minutos. Repetir solicitações `SendMessage` após a expiração do intervalo da eliminação de duplicação pode introduzir mensagens duplicadas na fila. Por exemplo, um dispositivo móvel em um carro envia mensagens cuja ordem é importante. Se o carro perder a conectividade celular por um período antes de receber uma confirmação, tentar novamente a solicitação depois de recuperada a conectividade celular pode criar uma duplicação.
- O consumidor deve ter um tempo limite de visibilidade que minimize o risco de não conseguir processar as mensagens antes que o tempo limite de visibilidade expire. Você pode estender o tempo limite de visibilidade enquanto as mensagens estão sendo processadas chamando a ação `ChangeMessageVisibility`. No entanto, se o tempo limite de visibilidade expirar, outro consumidor poderá começar imediatamente a processar as mensagens, fazendo com que uma mensagem seja processada várias vezes. Para evitar essa situação, configure uma dead letter queue.

Como trabalhar com tempos limite de visibilidade

Para um desempenho ideal, defina o tempo limite de visibilidade para ser maior do que o tempo limite de leitura do AWS SDK. Isso se aplica ao uso da ação de API `ReceiveMessage` com sondagem curta ou sondagem longa.

Usar o ID do grupo de mensagens do Amazon SQS

`MessageGroupId` é a etiqueta que especifica que uma mensagem pertence a um grupo de mensagens específico. As mensagens que pertencem ao mesmo grupo de mensagens são sempre processadas uma a uma, em uma ordem estrita relativa ao grupo de mensagens (no entanto, as mensagens que pertencem a diferentes grupos de mensagens podem ser processadas fora de ordem).

Intercalar vários grupos de mensagens ordenadas

Para intercalar vários grupos de mensagens ordenadas em uma única fila FIFO, use valores de IDs de grupos de mensagens (por exemplo, os dados de sessão de vários usuários). Nesse cenário, vários consumidores podem processar a fila, mas os dados de sessão de cada usuário são processados em uma forma FIFO.

> **Note**
>
> Quando as mensagens que pertencem a um determinado ID de grupo de mensagens são invisíveis, nenhum outro consumidor pode processar mensagens com o mesmo ID de grupo de mensagens.

Evitar o processamento de duplicações em um sistema de vários produtores/consumidores

Para evitar o processamento de mensagens duplicadas em um sistema com vários produtores e consumidores em que a taxa de transferência e latência são mais importantes do que a ordenação, o produtor deve gerar um ID de grupo de mensagens exclusivo para cada mensagem.

> **Note**
>
> Nesse cenário, duplicações são eliminadas. No entanto, a ordem da mensagem não pode ser garantida.

Qualquer cenário com vários produtores e consumidores aumenta o risco de entregar acidentalmente uma mensagem duplicada se um operador não processa a mensagem dentro do tempo limite de visibilidade e a mensagem se torna disponível para outro operador.

Evitar ter um grande backlog de mensagens com o mesmo ID de grupo de mensagens

Para filas FIFO, pode haver no máximo de 20.000 mensagens em trânsito (recebidas de uma fila por um consumidor, mas ainda não excluídas da fila). Se você atingir essa cota, o Amazon SQS não retornará nenhuma mensagem de erro. Uma fila FIFO examina as primeiras 20.000 mensagens para determinar os grupos de mensagens disponíveis. Isso significa que, se você tiver uma lista de pendências de mensagens em um único grupo de mensagens, não será possível consumir mensagens de outros grupos de mensagens que foram enviadas para a fila posteriormente até que você consuma com êxito as mensagens da lista de pendências.

Note

Um backlog de mensagens que têm o mesmo ID de grupo de mensagens poderá ser criado devido a um consumidor que não consegue processar uma mensagem com êxito. Podem ocorrer problemas de processamento de mensagens devido a um problema com o conteúdo de uma mensagem ou devido a um problema técnico com o consumidor.
Para remover mensagens que não podem ser processadas repetidamente e desbloquear o processamento de outras mensagens que têm o mesmo ID de grupo de mensagens, considere configurar uma política de dead-letter queue.

Evitar reutilizar o mesmo ID de grupo de mensagens com filas virtuais

Para impedir que mensagens com o mesmo ID de grupo de mensagens enviadas para filas virtuais diferentes com a mesma fila de host bloqueiem umas às outras, evite reutilizar o mesmo ID de grupo de mensagens com filas virtuais.

Usar o ID de tentativa de solicitação de recebimento do Amazon SQS

O ID de tentativa de solicitação de recebimento é o token usado para eliminação de duplicação de chamadas `ReceiveMessage`.

Durante uma interrupção de comunicação prolongada com a rede, que causa problemas de conectividade entre o SDK e o Amazon SQS, uma prática recomendada é fornecer o ID de tentativa

de solicitação de recebimento e tentar novamente, com o mesmo ID de tentativa de solicitação de recebimento, se a operação do SDK falhar.

Exemplos de SDK do Java do Amazon SQS

Você pode usar o AWS SDK for Java para criar aplicativos Java que interagem com o Amazon Simple Queue Service (Amazon SQS) e outros serviços. AWS Para instalar e configurar o SDK, consulte Conceitos básicos no Guia do desenvolvedor do AWS SDK for Java 2.x .

Para obter exemplos de operações básicas de fila do Amazon SQS, como criar uma fila ou enviar uma mensagem, consulte Trabalhar com filas de mensagens do Amazon SQS no Guia do desenvolvedor do AWS SDK for Java 2.x .

Os exemplos neste tópico demonstram recursos adicionais do Amazon SQS, como criptografia no lado do servidor (SSE), tags de alocação de custo e atributos de mensagem.

Tópicos

- Usando criptografia do lado do servidor com filas do Amazon SQS
- Configurando tags para uma fila do Amazon SQS
- Envio de atributos de mensagem para uma fila do Amazon SQS

Usando criptografia do lado do servidor com filas do Amazon SQS

Você pode usar o AWS SDK for Java para adicionar criptografia do lado do servidor (SSE) a uma fila do Amazon SQS. Cada fila usa uma chave KMS AWS Key Management Service (AWS KMS) para gerar as chaves de criptografia de dados. Este exemplo usa a chave KMS AWS gerenciada para o Amazon SQS. Para obter mais informações sobre como usar a SSE e a função da chave do KMS, consulte Criptografia em repouso no Amazon SQS.

Adicionar SSE a uma fila existente

Para habilitar a criptografia no lado do servidor para uma fila existente, use o método `SetQueueAttributes` para definir o atributo `KmsMasterKeyId`.

O exemplo de código a seguir define a AWS KMS key como a chave KMS AWS gerenciada para o Amazon SQS. O exemplo também define o período de reutilização de AWS KMS key como 140 segundos.

Antes de executar o código de exemplo, verifique se você definiu suas AWS credenciais. Para obter mais informações, consulte Configurar AWS credenciais e região para desenvolvimento no Guia do AWS SDK for Java 2.x desenvolvedor.

```
// 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);
```

Desabilitar a SSE para uma fila

Para desabilitar a criptografia no lado do servidor para uma fila existente, defina o atributo `KmsMasterKeyId` como uma string vazia usando o método `SetQueueAttributes`.

> **Important**
>
> `null` não é um valor válido para `KmsMasterKeyId`.

Criar uma fila com SSE

Para habilitar a SSE ao criar a fila, adicione o atributo `KmsMasterKeyId` ao método da API `CreateQueue`.

O exemplo a seguir cria uma fila nova com a SSE habilitada. A fila usa a chave do KMS gerenciada pela AWS para o Amazon SQS. O exemplo também define o período de reutilização de AWS KMS key como 160 segundos.

Antes de executar o código de exemplo, verifique se você definiu suas AWS credenciais. Para obter mais informações, consulte Configurar AWS credenciais e região para desenvolvimento no Guia do AWS SDK for Java 2.x desenvolvedor.

```
// 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);
```

Recuperar atributos de SSE

Para obter informações sobre como recuperar atributos da fila, consulte Exemplos na Referência da API do Amazon Simple Queue Service.

Para recuperar o ID da chave do KMS ou o período de reutilização da chave de dados de uma fila específica, execute o método `GetQueueAttributes` e recupere os valores `KmsMasterKeyId` e `KmsDataKeyReusePeriodSeconds`.

Configurando tags para uma fila do Amazon SQS

Use tags de alocação de custos em suas filas do Amazon SQS para ajudar a organizá-las e identificá-las. Os exemplos a seguir mostram como gerenciar tags usando o AWS SDK for Java. Para ter mais informações, consulte Tags de alocação de custos do Amazon SQS.

Antes de executar o código de exemplo, verifique se você definiu suas AWS credenciais. Para obter mais informações, consulte Configurar AWS credenciais e região para desenvolvimento no Guia do AWS SDK for Java 2.x desenvolvedor.

Listar tags

Para listar as tags de uma fila, use o 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() ));
```

Adicionar ou atualizar tags

Para adicionar ou atualizar valores de etiquetas em uma fila, use o 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);
```

Remover tags

Para remover uma ou mais tags da fila, use o método `UntagQueue`. O exemplo a seguir remove a tag `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);
```

Envio de atributos de mensagem para uma fila do Amazon SQS

É possível incluir metadados estruturados (como carimbos de data e hora, dados geoespaciais, assinaturas e identificadores) com mensagens usando os atributos de mensagem. Para ter mais informações, consulte Atributos de mensagem do Amazon SQS.

Antes de executar o código de exemplo, verifique se você definiu suas AWS credenciais. Para obter mais informações, consulte Configurar AWS credenciais e região para desenvolvimento no Guia do AWS SDK for Java 2.x desenvolvedor.

Definir atributos

Para definir um atributo para uma mensagem, adicione o código a seguir que usa o tipo de dado `MessageAttributeValue`. Para obter mais informações, consulte Componentes de atributos de mensagem e Tipos de dados de atributos de mensagem.

O calcula AWS SDK for Java automaticamente as somas de verificação do corpo da mensagem e do atributo da mensagem e as compara com os dados que o Amazon SQS retorna. Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for Java 2.x e Cálculo do resumo de mensagens MD5 para atributos de mensagem para outras linguagens de programação.

String

Este exemplo define um atributo `String` chamado `Name` com o valor `Jane`.

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

Number

Este exemplo define um atributo `Number` chamado `AccurateWeight` com o valor `230.00000000000000001`.

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

Binary

Este exemplo define um atributo `Binary` chamado `ByteArray` com o valor de uma matriz de 10 bytes não inicializada.

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

String (custom)

Este exemplo define o atributo personalizado `String.EmployeeId` chamado `EmployeeId` com o valor `ABC123456`.

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

Number (custom)

Este exemplo define o atributo personalizado `Number.AccountId` chamado `AccountId` com o valor `000123456`.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("AccountId", new MessageAttributeValue()
.withDataType("Number.AccountId")
.withStringValue("000123456"));
```

> **Note**
>
> Como o tipo de dados base é `Number`, o método `ReceiveMessage` retorna `123456`.

Binary (custom)

Este exemplo define um atributo personalizado `Binary.JPEG` chamado `ApplicationIcon` com o valor de uma matriz de 10 bytes não inicializada.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
```

```
messageAttributes.put("ApplicationIcon", new MessageAttributeValue()
.withDataType("Binary.JPEG")
.withBinaryValue(ByteBuffer.wrap(new byte[10])));
```

Enviar uma mensagem com atributos

Este exemplo adiciona os atributos à `SendMessageRequest` antes de enviar a mensagem.

```
// 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

Se você enviar uma mensagem para uma fila primeiro a entrar, primeiro a sair (FIFO), verifique se o método `sendMessage` é executado depois que você fornecer o ID do grupo de mensagens.

Se usar o método `SendMessageBatch` em vez de `SendMessage`, você deverá especificar os atributos da mensagem de cada mensagem no lote.

Trabalhar com APIs do Amazon SQS

Esta seção fornece informações sobre como criar endpoints do Amazon SQS, fazer solicitações da API de consulta com os métodos GET e POST e usar ações em lote da API. Para ter informações detalhadas sobre as ações do Amazon SQS, inclusive parâmetros, erros, exemplos e tipos de dados, consulte a Referência da API do Amazon Simple Queue Service.

Para acessar o Amazon SQS utilizando várias linguagens de programação, você também pode usar os AWS SDKs que contêm a seguinte funcionalidade automática:

- Assinar criptograficamente suas solicitações de serviço
- Recuperar solicitações
- Lidar com respostas de erro

Para ter informações sobre a ferramenta da linha de comando, consulte as seções do Amazon SQS na Referência de comandos da AWS CLI e na Referência de Cmdlet do AWS Tools for PowerShell .

APIs do Amazon SQS com protocolo JSON AWS

O Amazon SQS usa o protocolo AWS JSON como mecanismo de transporte para todas as APIs do Amazon SQS nas versões especificadas do SDK.AWS AWS O protocolo JSON fornece maior taxa de transferência, menor latência e comunicação mais rápida. application-to-application AWS O protocolo JSON é mais eficiente na serialização/desserialização de solicitações e respostas quando comparado ao protocolo de consulta. AWS Se você ainda preferir usar o protocolo de AWS consulta com as APIs do SQS, consulte Quais linguagens são compatíveis com o protocolo JSON da AWS usado nas APIs do Amazon SQS? as versões do AWS SDK que suportam o protocolo de consulta do Amazon SQS. AWS

O Amazon SQS usa o protocolo AWS JSON para se comunicar entre clientes AWS SDK (por exemplo, Java, Python, Golang) e JavaScript o servidor Amazon SQS. Uma solicitação HTTP de uma operação de API do Amazon SQS aceita entrada formatada em JSON. A operação do Amazon SQS é executada, e a resposta de execução é enviada de volta ao cliente do SDK no formato JSON. Comparado à AWS consulta, o AWS JSON é mais simples, rápido e eficiente para transportar dados entre cliente e servidor.

- AWS O protocolo JSON atua como um mediador entre o cliente e o servidor Amazon SQS.

- O servidor não entende a linguagem de programação na qual a operação do Amazon SQS é criada, mas entende o protocolo AWS JSON.
- O protocolo AWS JSON usa a serialização (converter objeto para o formato JSON) e a desserialização (converter formato JSON em objeto) entre o cliente e o servidor do Amazon SQS.

Para obter mais informações sobre o protocolo AWS JSON com o Amazon SQS, consulte. Perguntas frequentes sobre o protocolo Amazon SQS AWS JSON

AWS O protocolo JSON está disponível na versão especificada do AWS SDK. Para examinar a versão do SDK e as datas de lançamento em todas as variantes de linguagem, consulte a Matriz de suporte a versões de AWS SDKs e ferramentas no Guia de referência de AWS SDKs e ferramentas.

Tópicos

- Fazer solicitações de API de consulta usando o protocolo AWS JSON no Amazon SQS
- Fazer solicitações de API de AWS consulta usando o protocolo de consulta no Amazon SQS
- Autenticação de solicitações para o Amazon SQS
- Ações em lote do Amazon SQS

Fazer solicitações de API de consulta usando o protocolo AWS JSON no Amazon SQS

Nesta seção, você aprenderá a criar um endpoint do Amazon SQS, fazer solicitações POST e interpretar respostas.

> **Note**
>
> AWS O protocolo JSON é compatível com a maioria das variantes de linguagem. Para ver uma lista completa de variantes de linguagem compatíveis, consulte Quais linguagens são compatíveis com o protocolo JSON da AWS usado nas APIs do Amazon SQS?.

Tópicos

- Criar um endpoint
- Como fazer uma solicitação POST

- Interpretar as respostas da API JSON do Amazon SQS
- Perguntas frequentes sobre o protocolo Amazon SQS AWS JSON

Criar um endpoint

Para trabalhar com filas do Amazon SQS, você deve criar um endpoint. Para ter informações sobre endpoints do Amazon SQS, consulte as seguintes páginas na Referência geral da Amazon Web Services:

- Endpoints regionais
- Endpoints e cotas do Amazon Simple Queue Service

Cada endpoint do Amazon SQS é totalmente independente. Por exemplo, se duas filas forem nomeadas MyQueuee uma tiver o endpoint `sqs.us-east-2.amazonaws.com` enquanto a outra tiver o endpoint`sqs.eu-west-2.amazonaws.com`, as duas filas não compartilharão dados entre si.

Veja a seguir um exemplo de endpoint que faz uma solicitação para criar uma fila.

```
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**
>
> Os nomes e os URLs de fila diferenciam maiúsculas e minúsculas.

A estrutura de *AUTHPARAMS* depende de como você assina sua solicitação de API. Para obter mais informações, consulte Assinar solicitações de AWS API na Referência geral da Amazon Web Services.

Como fazer uma solicitação POST

As solicitações POST do Amazon SQS enviam parâmetros de consulta como um formulário no corpo de uma solicitação HTTP.

Veja a seguir um exemplo de cabeçalho HTTP com `X-Amz-Target` definido como `AmazonSQS.<operationName>` e um cabeçalho HTTP com `Content-Type` definido como `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",
}
```

Essa solicitação HTTP POST envia uma mensagem a uma fila do Amazon SQS.

Note

Os dois cabeçalhos HTTP `X-Amz-Target` e `Content-Type` são obrigatórios.
Seu cliente HTTP pode adicionar outros itens à solicitação HTTP, de acordo com a versão do HTTP do cliente.

Interpretar as respostas da API JSON do Amazon SQS

Em resposta a uma solicitação de ação, o Amazon SQS retorna uma estrutura de dados JSON que contém os resultados da solicitação. Para receber mais informações, consulte ações individuais

na Referência da API do Amazon Simple Queue Service e Perguntas frequentes sobre o protocolo Amazon SQS AWS JSON.

Tópicos

Estrutura de resposta JSON bem-sucedida

Se a solicitação for bem-sucedida, o principal elemento de resposta será `x-amzn-RequestId`, que contém o identificador único universal (UUID) da solicitação, bem como outros campos de resposta anexados. Por exemplo, a resposta `CreateQueue` contém o campo `QueueUrl`, que, por sua vez, contém o URL da fila criada.

```
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"
}
```

Estrutura de resposta de erro JSON

Se uma solicitação não for bem-sucedida, o Amazon SQS retornará a resposta principal, incluindo o cabeçalho HTTP e o corpo.

No cabeçalho HTTP, `x-amzn-RequestId` contém o UUID da solicitação. `x-amzn-query-error` contém duas informações: o tipo de erro e se o erro foi do produtor ou do consumidor.

No corpo de resposta, `"__type"` indica outros detalhes do erro, e `Message` indica a condição de erro em um formato legível.

Veja a seguir um exemplo de resposta de erro no 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."
}
```

Perguntas frequentes sobre o protocolo Amazon SQS AWS JSON

Perguntas frequentes sobre o uso do protocolo AWS JSON com o Amazon SQS.

O que é o protocolo AWS JSON e como ele difere das solicitações e respostas existentes da API Amazon SQS?

JSON é um dos métodos de conexão mais amplamente usados e aceitos para comunicação entre sistemas heterogêneos. O Amazon SQS usa o JSON como meio de comunicação entre um cliente AWS SDK (por exemplo, Java, Python, Golang) JavaScript e o servidor Amazon SQS. Uma solicitação HTTP de uma operação de API do Amazon SQS aceita entrada formatada em JSON. A operação do Amazon SQS é executada, e a resposta de execução é compartilhada de volta com o cliente do SDK no formato JSON. Comparado com a consulta da AWS , o JSON é mais eficiente para transportar dados entre o cliente e o servidor.

- O protocolo Amazon SQS AWS JSON atua como um mediador entre o cliente e o servidor do Amazon SQS.
- O servidor não entende a linguagem de programação na qual a operação do Amazon SQS é criada, mas entende o protocolo AWS JSON.
- O protocolo Amazon SQS AWS JSON usa a serialização (converter objeto no formato JSON) e a desserialização (converter formato JSON em objeto) entre o cliente e o servidor do Amazon SQS.

Como faço para começar a usar os protocolos AWS JSON para o Amazon SQS?

Para começar com a versão mais recente do AWS SDK e obter mensagens mais rápidas para o Amazon SQS, atualize AWS seu SDK para a versão especificada ou qualquer versão posterior. Para saber mais sobre os clientes do SDK, consulte a coluna “Guia” na tabela abaixo.

A seguir está uma lista de versões do SDK em todas as variantes de linguagem do protocolo AWS JSON para uso com as APIs do Amazon SQS:

Idioma	Repositório do cliente do SDK	Versão obrigatória do cliente do SDK	Guia
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 em AWS
JavaScript v3.x	aws/aws-sdk-js-v3	v3.447.0	JavaScript em 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)
Python-botocore	boto/botocore	1.31.82	AWS SDK para Python (Boto3)
awscli	AWS CLI	1.29.82	AWS Command Line Interface

Idioma	Repositório do cliente do SDK	Versão obrigatória do cliente do SDK	Guia
Ruby	aws/aws-sdk-ruby	1.67,0	AWS SDK para Ruby

Quais são os riscos de habilitar o protocolo JSON para minhas workloads do Amazon SQS?

Se você estiver usando uma implementação personalizada do AWS SDK ou uma combinação de clientes personalizados e AWS SDK para interagir com o Amazon SQS que AWS gera respostas baseadas em consultas (também conhecidas como baseadas em XML), ela pode ser incompatível com o protocolo JSON. AWS Se você encontrar algum problema, entre em contato com o AWS Support.

E se eu já estiver usando a versão mais recente do AWS SDK, mas minha solução de código aberto não for compatível com JSON?

É necessário alterar a versão do SDK para a versão anterior à que você está usando. Consulte Como faço para começar a usar os protocolos AWS JSON para o Amazon SQS? para obter mais informações. AWS As versões do SDK listadas em Como faço para começar a usar os protocolos AWS JSON para o Amazon SQS? usam o protocolo JSON wire para as APIs do Amazon SQS. Se você alterar seu AWS SDK para a versão anterior, suas APIs do Amazon SQS usarão a consulta. AWS

Quais linguagens são compatíveis com o protocolo JSON da AWS usado nas APIs do Amazon SQS?

O Amazon SQS oferece suporte a todas as variantes de idioma nas quais os AWS SDKs estão geralmente disponíveis (GA). No momento, não há compatibilidade com Kotlin, Rust e Swift. Para saber mais sobre outras variantes de linguagem, consulte Ferramentas para criar com a AWS.

Quais regiões são compatíveis com o protocolo JSON da AWS usado nas APIs do Amazon SQS?

O Amazon SQS oferece suporte ao protocolo AWS JSON em todas as AWS regiões em que o Amazon SQS está disponível.

Quais melhorias de latência posso esperar ao atualizar para as versões especificadas do AWS SDK para o Amazon SQS usando o protocolo JSON? AWS

AWS O protocolo JSON é mais eficiente na serialização e desserialização de solicitações e respostas quando comparado ao protocolo de consulta. AWS Com base em testes de AWS desempenho para uma carga útil de mensagem de 5 KB, o protocolo JSON para Amazon SQS end-to-end reduz a latência do processamento de mensagens em até 23% e reduz o uso da CPU e da memória do lado do cliente do aplicativo.

O protocolo AWS de consulta será descontinuado?

AWS o protocolo de consulta continuará sendo suportado. Você pode continuar usando o protocolo de AWS consulta, desde que sua versão do AWS SDK esteja definida como uma versão anterior diferente da listada em Como começar a usar protocolos AWS JSON para Amazon SQS.

Onde posso receber mais informações sobre o protocolo JSON da AWS ?

Você pode receber mais informações sobre o protocolo JSON emAWS JSON 1.0 protocol na documentação da Smithy. Para saber mais sobre as solicitações de API do Amazon SQS usando o protocolo JSON da AWS , consulte Fazer solicitações de API de consulta usando o protocolo AWS JSON no Amazon SQS.

Fazer solicitações de API de AWS consulta usando o protocolo de consulta no Amazon SQS

Nesta seção, você aprenderá a criar um endpoint do Amazon SQS, fazer solicitações GET e POST e a interpretar respostas.

Tópicos

- Criar um endpoint
- Como fazer uma solicitação GET
- Como fazer uma solicitação POST
- Interpretar as respostas da API XML do Amazon SQS

Criar um endpoint

Para trabalhar com filas do Amazon SQS, você deve criar um endpoint. Para ter informações sobre endpoints do Amazon SQS, consulte as seguintes páginas na Referência geral da Amazon Web Services:

- Endpoints regionais
- Endpoints e cotas do Amazon Simple Queue Service

Cada endpoint do Amazon SQS é totalmente independente. Por exemplo, se duas filas forem nomeadas MyQueuee uma tiver o endpoint `sqs.us-east-2.amazonaws.com` enquanto a outra tiver o endpoint`sqs.eu-west-2.amazonaws.com`, as duas filas não compartilharão dados entre si.

Veja a seguir um exemplo de um endpoint que faz uma solicitação para criar uma fila.

```
https://sqs.eu-west-2.amazonaws.com/
?Action=CreateQueue
&DefaultVisibilityTimeout=40
&QueueName=MyQueue
&Version=2012-11-05
&AUTHPARAMS
```

> **Note**
>
> Os nomes e os URLs de fila diferenciam maiúsculas e minúsculas.
> A estrutura de *AUTHPARAMS* depende de como você assina sua solicitação de API. Para obter mais informações, consulte Assinar solicitações de AWS API na Referência geral da Amazon Web Services.

Como fazer uma solicitação GET

Uma solicitação GET do Amazon SQS é estruturada como um URL que consiste no seguinte:

- Endpoint: o recurso no qual a solicitação está agindo (o nome da fila e o URL), por exemplo: `https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue`
- Ação: a ação que você quer executar no endpoint. Um ponto de interrogação (?) separa o endpoint da ação, por exemplo: `?Action=SendMessage&MessageBody=Your%20Message%20Text`

- Parâmetros: os parâmetros da solicitação. Cada parâmetro é separado por um E comercial (&); por exemplo: `&Version=2012-11-05&`*`AUTHPARAMS`*

Veja a seguir um exemplo de solicitação GET que envia mensagens a uma fila do Amazon SQS.

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue
?Action=SendMessage&MessageBody=Your%20message%20text
&Version=2012-11-05
&AUTHPARAMS
```

> **Note**
>
> Os nomes e os URLs de fila diferenciam maiúsculas e minúsculas.
> Como as solicitações GET são URLs, você deve codificar todos os valores de parâmetro no URL. Como não são permitidos espaços nos URLs, cada espaço é codificado no URL como `%20`. O restante do exemplo não foi codificado no URL para facilitar a leitura.

Como fazer uma solicitação POST

As solicitações POST do Amazon SQS enviam parâmetros de consulta como um formulário no corpo de uma solicitação HTTP.

Veja a seguir um exemplo de cabeçalho HTTP com `Content-Type` definido como `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
```

O cabeçalho é seguido por uma solicitação GET `form-urlencoded` que envia uma mensagem a uma fila do Amazon SQS. Cada parâmetro é separado por um E comercial (&).

```
Action=SendMessage
&MessageBody=Your+Message+Text
&Expires=2020-10-15T12%3A00%3A00Z
&Version=2012-11-05
&AUTHPARAMS
```

> **Note**
>
> Somente o cabeçalho HTTP `Content-Type` é obrigatório. O *`AUTHPARAMS`* é o mesmo para a solicitação GET.
> Seu cliente HTTP pode adicionar outros itens à solicitação HTTP, de acordo com a versão do HTTP do cliente.

Interpretar as respostas da API XML do Amazon SQS

Em resposta a uma solicitação de ação, o Amazon SQS retorna uma estrutura de dados XML que contém os resultados da solicitação. Para obter mais informações, consulte ações individuais na Referência da API do Amazon Simple Queue Service.

Tópicos

- Estrutura de resposta de XML bem-sucedida
- Estrutura de resposta de erro de XML

Estrutura de resposta de XML bem-sucedida

Se a solicitação for bem-sucedida, o elemento de resposta principal receberá o nome da ação, com `Response` anexada (por exemplo, *`ActionName`*`Response`).

Esse elemento contém os seguintes elementos filho:

- ***`ActionName`*`Result`**: contém um elemento específico à ação. Por exemplo, o elemento `CreateQueueResult` contém o elemento `QueueUrl` que, por sua vez, contém o URL da fila criada.
- **`ResponseMetadata`**: contém o `RequestId`, que, por sua vez, contém o Universal Unique Identifier (UUID) da solicitação.

Veja a seguir um exemplo de uma resposta bem-sucedida no 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>
```

Estrutura de resposta de erro de XML

Se uma solicitação não tiver êxito, o Amazon SQS retornará o elemento de resposta principal `ErrorResponse`. Esse elemento contém um elemento `Error` e um elemento `RequestId`.

O elemento `Error` contém os seguintes elementos filhos:

- **`Type`**: especifica se o erro foi de um produtor ou de um consumidor.
- **`Code`**: especifica o tipo de erro.
- **`Message`**: especifica a condição do erro em um formato legível.
- **`Detail`**: (opcional) especifica detalhes adicionais sobre o erro.

O elemento `RequestId` contém o UUID do pedido.

Veja a seguir um exemplo de uma resposta com erro no 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>
```

Autenticação de solicitações para o Amazon SQS

A autenticação é o processo para identificar e verificar quem envia uma solicitação. Durante a primeira etapa de autenticação, a AWS verifica a identidade do produtor e se ele está registrado para

usar a AWS (para obter mais informações, consulte Etapa 1: criar um usuário Conta da AWS e IAM). Em seguida, AWS segue o seguinte procedimento:

1. O produtor (remetente) obtém as credenciais necessárias.
2. O produtor envia uma solicitação e a credencial para o consumidor (destinatário).
3. O consumidor usa a credencial para verificar se o produtor enviou a solicitação.
4. Uma das seguintes situações acontece:
   - Se a autenticação for bem-sucedida, o consumidor processará a solicitação.
   - Se a autenticação falhar, o consumidor rejeitará a solicitação e retornará um erro.

Tópicos

Processo de autenticação básica com HMAC-SHA

Ao acessar o Amazon SQS usando a API de consulta, você deve fornecer os seguintes itens para que a solicitação seja autenticada:

- O ID da chave de AWS acesso que identifica sua Conta da AWS, que é AWS usado para pesquisar sua chave de acesso secreta.
- A assinatura da solicitação HMAC-SHA, que é calculada usando sua chave de acesso secreta (um segredo compartilhado do qual somente você e a AWS têm conhecimento. Para obter mais informações, consulte RFC2104). O SDK da AWS lida com o processo de assinatura. No entanto, se você enviar uma solicitação de consulta por meio de HTTP ou HTTPS, precisará incluir uma assinatura em cada solicitação de consulta.

  1. Derive uma chave de assinatura Signature versão 4. Para obter mais informações, consulte Derivar a chave de assinatura com Java.

     > **Note**
     >
     > O Amazon SQS oferece suporte ao Signature versão 4, que fornece segurança e performance aprimorados com base em SHA256 em relação às versões anteriores.

Ao criar novas aplicações que usem o Amazon SQS, você deverá usar o Signature versão 4.

2. Codifique a assinatura da solicitação usando base64. Este exemplo do código Java faz o seguinte:

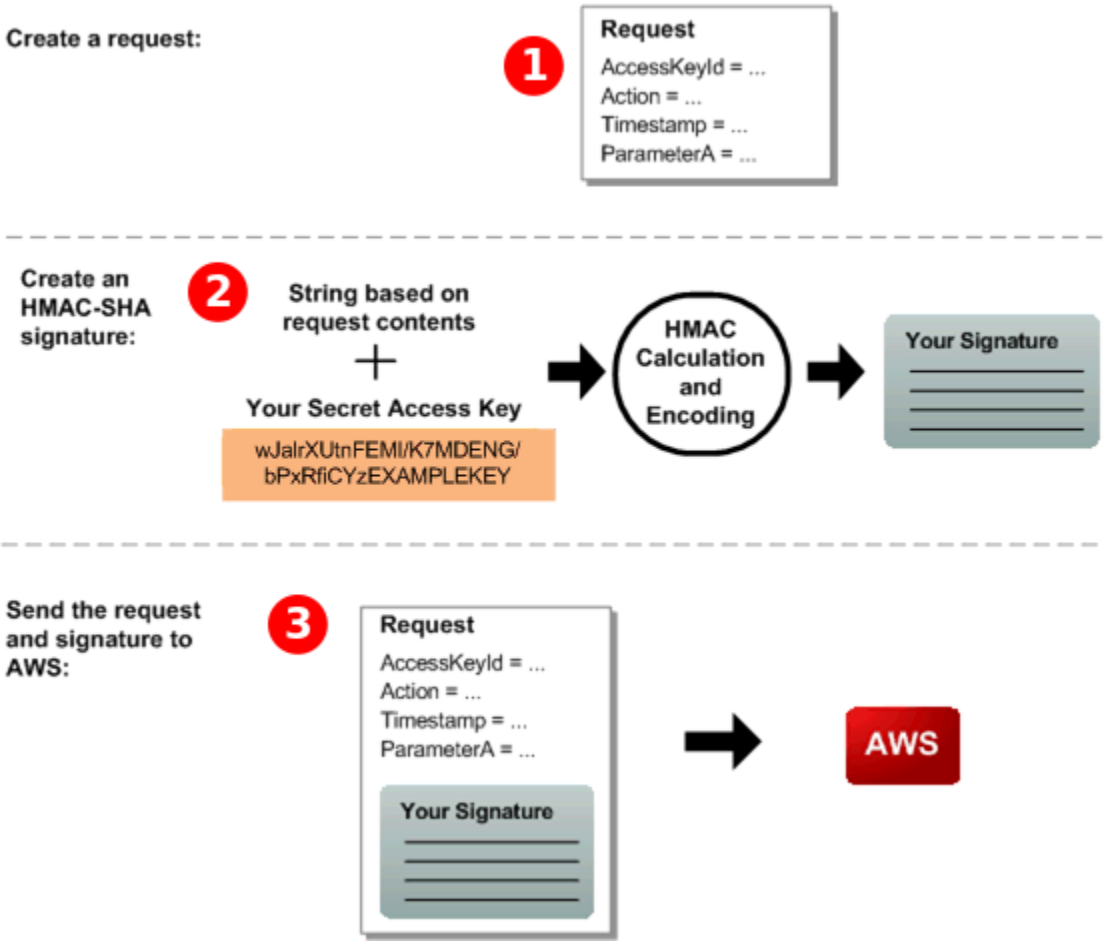
```
package amazon.webservices.common;

// 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);
    }
}
```

- O timestamp (ou a expiração) da solicitação. O timestamp que você usa na solicitação deve ser um objeto `dateTime`, com a data completa, incluindo horas, minutos e segundos. Por exemplo: `2007-01-31T23:59:59Z` Embora isso não seja necessário, recomendamos que você informe o objeto usando o fuso horário Tempo Universal Coordenado (Horário de Greenwich).

> **Note**
>
> Certifique-se de que a hora do servidor esteja definida corretamente. Se você especificar um registro de data e hora (em vez de uma expiração), a solicitação expirará automaticamente 15 minutos após o horário especificado (AWS não processará solicitações com carimbos de data e hora mais de 15 minutos antes da hora atual nos servidores). AWS
> Se você está usando .NET, não deve enviar timestamps excessivamente específicos (devido a interpretações diferentes em relação a como a precisão de tempo extra deve ser aplicada). Neste caso, você deve criar objetos `dateTime` manualmente com precisão inferior a um milissegundo.

Parte 1: a solicitação do usuário

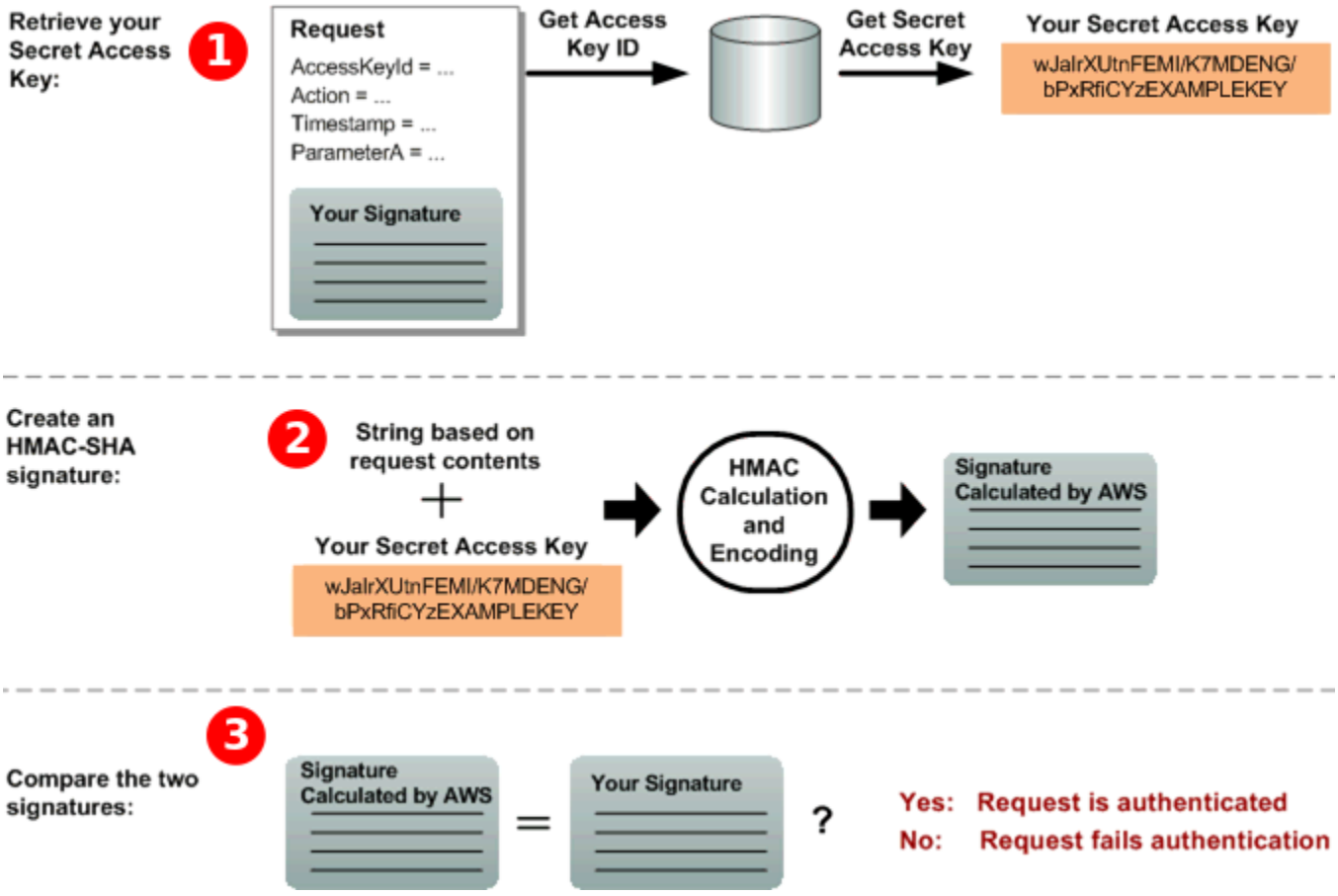
A seguir está o processo que você deve seguir para autenticar AWS solicitações usando uma assinatura de solicitação HMAC-SHA.



1. Crie uma solicitação para AWS.
2. Calcule uma assinatura com código de autenticação de mensagem de hash com chave (HMAC-SHA) usando sua chave de acesso secreta.
3. Inclua a assinatura e o ID da chave de acesso na solicitação e, em seguida, envie a solicitação para AWS.

Parte 2: A resposta de AWS

AWS inicia o seguinte processo em resposta.



1. AWS usa o ID da chave de acesso para pesquisar sua chave de acesso secreta.
2. AWS gera uma assinatura a partir dos dados da solicitação e da chave de acesso secreta, usando o mesmo algoritmo usado para calcular a assinatura enviada na solicitação.
3. Uma das seguintes situações acontece:
   - Se a assinatura AWS gerada corresponder à que você enviou na solicitação, AWS considere a solicitação autêntica.
   - Se a comparação falhar, a solicitação será descartada e AWS retornará um erro.

Ações em lote do Amazon SQS

Para reduzir custos ou para manipular até 10 mensagens com uma única ação, você pode usar as seguintes ações:

- `SendMessageBatch`
- `DeleteMessageBatch`

- `ChangeMessageVisibilityBatch`

Você pode aproveitar a funcionalidade em lote usando a API de consulta ou um AWS SDK que suporte as ações em lote do Amazon SQS.

> **Note**
>
> O tamanho total de todas as mensagens enviadas em uma única `SendMessageBatch` chamada não pode exceder 262.144 bytes (256 KiB).
> Não é possível definir permissões para `SendMessageBatch`, `DeleteMessageBatch` ou `ChangeMessageVisibilityBatch` explicitamente. A definição de permissões para `SendMessage`, `DeleteMessage` ou `ChangeMessageVisibility` define permissões para as versões de lote correspondentes dessas ações.
> O console do Amazon SQS não oferece suporte a ações em lote.

Tópicos

Habilitando o buffer do lado do cliente e o agrupamento de solicitações com o Amazon SQS

O AWS SDK for Java inclui o `AmazonSQSBufferedAsyncClient`, que acessa o Amazon SQS. Esse cliente permite o envio simples de solicitações em lotes usando armazenamento em buffer no lado do cliente. As chamadas feitas pelo cliente são primeiro armazenadas em buffer e, em seguida, enviadas como uma solicitação em lote para o Amazon SQS.

O armazenamento em buffer no lado do cliente permite que até 10 solicitações sejam armazenadas em buffer e enviadas como uma solicitação em lote, diminuindo o custo de uso do Amazon SQS e reduzindo o número de solicitações enviadas. O `AmazonSQSBufferedAsyncClient` armazena tanto as chamadas síncronas quanto as assíncronas em buffer. Solicitações em lote e suporte para sondagem longa também podem ajudar a aumentar a taxa de transferência. Para ter mais informações, consulte Aumento da produtividade usando escalabilidade horizontal e lotes de ações com o Amazon SQS.

como o `AmazonSQSBufferedAsyncClient` implementa a mesma interface que o `AmazonSQSAsyncClient`, migrar de `AmazonSQSAsyncClient` para `AmazonSQSBufferedAsyncClient` normalmente requer apenas pequenas mudanças no seu código existente.

> **Note**
>
> Atualmente, o cliente assíncrono no buffer do Amazon SQS não oferece suporte a filas FIFO.

Tópicos

Usando o cliente BufferedAsync AmazonSQS

Antes de começar, conclua as etapas em Configurar o Amazon SQS.

> **Important**
>
> No momento, o AWS SDK for Java 2.x não é compatível com `AmazonSQSBufferedAsyncClient` o.

Você pode criar um novo `AmazonSQSBufferedAsyncClient` com base em `AmazonSQSAsyncClient`, por exemplo:

```
// Create the basic Amazon SQS async client
final AmazonSQSAsync sqsAsync = new AmazonSQSAsyncClient();

// Create the buffered client
final AmazonSQSAsync bufferedSqs = new AmazonSQSBufferedAsyncClient(sqsAsync);
```

Depois de criar o novo `AmazonSQSBufferedAsyncClient`, você pode usá-lo para enviar várias solicitações ao Amazon SQS (da mesma forma que faria com o `AmazonSQSAsyncClient`), por exemplo:

```
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);
```

Configurando o cliente BufferedAsync AmazonSQS

O `AmazonSQSBufferedAsyncClient` é pré-configurado com configurações que funcionarão para a maioria dos casos de uso. Você pode configurar ainda mais o `AmazonSQSBufferedAsyncClient`, por exemplo:

1. Crie uma instância da classe `QueueBufferConfig` com os parâmetros de configuração necessários.
2. Informe a instância para o construtor `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 parâmetros de configuração

Parâmetro	Valor padrão	Descrição
`longPoll`	`true`	

Parâmetro	Valor padrão	Descrição
		Quando `longPoll` está definido como `true`, `AmazonSQSBufferedAsyncClient` tenta usar a sondagem longa ao consumir mensagens.
`longPollWaitTimeoutSeconds`	20 s	A quantidade máxima de tempo, em segundos, em que uma chamada `ReceiveMessage` é bloqueada no servidor aguardando as mensagens aparecerem na fila antes de retornar com um resultado de recebimento vazio. **Note** Quando a sondagem longa está desativada, essa configuração não tem efeito.

Parâmetro	Valor padrão	Descrição
`maxBatchOpenMs`	200ms	A quantidade máxima de tempo (em milissegundos) que uma chamada de saída aguarda outras chamadas com as quais ela coloca mensagens do mesmo tipo em lote. Quanto maior for a configuração, menos lotes serão necessários para executar a mesma quantidade de trabalho (no entanto, a primeira chamada em um lote deve passar mais tempo em espera). Quando esse parâmetro é definido como `0`, as solicitações enviadas não aguardam outras solicitações, desativando efetivamente o processamento em lotes.

Parâmetro	Valor padrão	Descrição
`maxBatchSize`	10 solicitações por lote	O número máximo de mensagens que são armazenadas em lote em uma única solicitação. Quanto maior a configuração, menos lotes serão necessários para executar o mesmo número de solicitações. **Note** Dez solicitações por lote é o valor máximo permitido para o Amazon SQS.
`maxBatchSizeBytes`	256 KiB	O tamanho máximo de um lote de mensagens, em bytes, que o cliente tenta enviar ao Amazon SQS. **Note** 256 KiB é o valor máximo permitido para o Amazon SQS.

Parâmetro	Valor padrão	Descrição
`maxDoneReceiveBatches`	10 lotes	O número máximo de lotes de recebimento que `AmazonSQSBufferedAsyncClient` pré-busca e armazena no lado do cliente. Quanto maior for a configuração, mais solicitações de recebimento poderão ser atendidas sem a necessidade de de fazer uma chamada ao Amazon SQS (no entanto, quanto mais mensagens forem buscadas previamente, mais tempo elas permanecerão no buffer, fazendo com que o tempo limite de visibilidade expire). **Note** `0`indica que toda a pré-busca de mensagens está desativada e que as mensagens são consumidas somente sob demanda.

Parâmetro	Valor padrão	Descrição
`maxInflightOutboundBatches`	5 lotes	O número máximo de lotes de saída ativos que podem ser processados ao mesmo tempo. Quanto maior for a configuração, mais rapidamente os lotes de saída poderão ser enviados (sujeito a outras cotas, como CPU ou largura de banda) e mais threads serão consumidos pelo `AmazonSQSBufferedAsyncClient` .

Parâmetro	Valor padrão	Descrição
`maxInflightReceiveBatches`	10 lotes	O número máximo de lotes de recebimento ativos que podem ser processados ao mesmo tempo. Quanto maior for a configuração, mais mensagens serão recebidas (sujeito a outras cotas, como CPU ou largura de banda) e mais threads serão consumidos pelo `AmazonSQSBufferedAsyncClient` . **Note** `0`indica que toda a pré-busca de mensagens está desativada e que as mensagens são consumidas somente sob demanda.

Parâmetro	Valor padrão	Descrição
`visibilityTimeoutSeconds`	-1	Quando esse parâmetro é definido como um valor positivo e diferente de zero, o tempo limite de visibilidade definido aqui substitui o tempo limite de visibilidade definido na fila a partir da qual as mensagens são consumidas. **Note** `-1` indica que a configuração padrão foi selecionada para a fila. Não é possível configurar o tempo limite de visibilidade para `0`.

Aumento da produtividade usando escalabilidade horizontal e lotes de ações com o Amazon SQS

As filas do Amazon SQS podem fornecer taxa de transferência muito altas. Para obter mais informações sobre cotas de taxa de transferência, consulte Cotas de mensagens do Amazon SQS.

Para atingir uma taxa de transferência alta, você deve dimensionar os produtores de mensagens e os consumidores horizontalmente (adicionar mais produtores e consumidores).

Tópicos

- Escalabilidade horizontal
- Processamento de ações em lotes

- Exemplo de Java funcional para operações únicas e solicitações em lote

Escalabilidade horizontal

Como você acessa o Amazon SQS por meio de um protocolo HTTP de solicitação-resposta, a latência da solicitação (o intervalo de tempo entre o início de uma solicitação e o recebimento de uma resposta) limita a taxa de transferência que você pode obter de uma única thread por meio de uma única conexão. Por exemplo, se a latência média de um cliente com base no Amazon EC2 para o Amazon SQS na mesma região for de cerca de 20 ms, a taxa de transferência máxima de uma única thread por meio de uma única conexão será em média 50 TPS.

A escalabilidade horizontal envolve o aumento do número de produtores de mensagem (que fazem a solicitação `SendMessage`) e dos consumidores (que fazem solicitações `ReceiveMessage` e `DeleteMessage`) para aumentar sua taxa de transferência de fila geral. Você pode escalar horizontalmente de três formas:

- Aumentar o número de threads por cliente
- Adicionar mais clientes
- Aumentar o número de threads por cliente e adicionar mais clientes

Ao adicionar mais clientes, você obtém ganhos essencialmente lineares na taxa de transferência da fila. Por exemplo, se você dobrar o número de clientes, terá duas vezes a taxa de transferência.

> **Note**
>
> À medida que você escala horizontalmente, é necessário garantir que o cliente do Amazon SQS tenha conexões ou threads suficientes para oferecer suporte à quantidade de produtores e consumidores de mensagens simultâneos que enviarão solicitações e receberão respostas. Por exemplo, por padrão, as instâncias da AWS SDK for Java `AmazonSQSClient` classe mantêm no máximo 50 conexões com o Amazon SQS. Para criar produtores e consumidores simultâneos adicionais, você precisa ajustar o número máximo de threads de produtores e consumidores permitidos em um objeto `AmazonSQSClientBuilder`, por exemplo:
>
> ```
> final AmazonSQS sqsClient = AmazonSQSClientBuilder.standard()
>         .withClientConfiguration(new ClientConfiguration()
>                 .withMaxConnections(producerCount + consumerCount))
> ```

```
        .build();
```

Para `AmazonSQSAsyncClient`, você também precisa ter certeza de que há threads suficientes disponíveis.
Esse exemplo só funciona para Java v. 1.x.

Processamento de ações em lotes

O processamento em lotes executa mais trabalho durante a ida e a volta do serviço (por exemplo, quando você envia várias mensagens com uma única solicitação `SendMessageBatch`). As ações de em lote do Amazon SQS são `SendMessageBatch`, `DeleteMessageBatch` e `ChangeMessageVisibilityBatch`. Para aproveitar o processamento em lotes sem alterar os produtores ou consumidores, você pode usar o cliente assíncrono armazenado em buffer para o Amazon SQS.

> **Note**
>
> Como `ReceiveMessage` pode processar 10 mensagens por vez, não há nenhuma ação `ReceiveMessageBatch`.

O processamento em lotes distribui a latência da ação de lote nas várias mensagens de uma solicitação em lote em vez de aceitar toda a latência para uma única mensagem (por exemplo, uma solicitação `SendMessage`). Como cada ida e volta carrega mais trabalho, as solicitações de lote tornam mais eficiente o uso de threads e conexões, melhorando, dessa forma, a taxa de transferência.

Você pode combinar processamentos em lote com escalabilidade horizontal para fornecer taxa de transferência com menos threads, conexões e solicitações em comparação com as solicitações de mensagens individuais. Você pode usar ações em lotes do Amazon SQS para enviar, receber ou excluir até 10 mensagens por vez. Como o Amazon SQS cobra por solicitação, o processamento em lotes pode reduzir substancialmente os custos.

O processamento em lotes pode criar certa complexidade para o seu aplicativo (por exemplo, o aplicativo precisa acumular as mensagens antes de enviá-las e, às vezes, precisará esperar mais por uma resposta). No entanto, o processamento em lotes pode ser eficaz nos seguintes casos:

- Seu aplicativo gera muitas mensagens em um curto intervalo de tempo, portanto, o atraso nunca é muito longo.
- Um consumidor de mensagem busca as mensagens de uma fila a seu critério, ao contrário de produtores de mensagem típicos que precisam enviar mensagens em resposta a eventos que eles não controlam.

> **Important**
>
> Uma solicitação de lote pode ser bem-sucedida, mesmo que ocorra falha nas mensagens individuais no lote. Após uma solicitação de lote, você sempre deve verificar a existência de falhas em mensagens individuais e repetir a ação, se necessário.

Exemplo de Java funcional para operações únicas e solicitações em lote

Pré-requisitos

Adicione os pacotes `aws-java-sdk-sqs.jar`, `aws-java-sdk-ec2.jar` e `commons-logging.jar` ao caminho da classe de compilação do Java. O exemplo a seguir mostra essas dependências em um arquivo `pom.xml` do projeto 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

O exemplo de código Java a seguir implementa um padrão simples de produtor-consumidor. O thread principal gera um número de threads de produtor e consumidor que processam mensagens de 1 KB em um determinado momento. Ele inclui produtores e os consumidores que fazem solicitações de operação únicas e outros que fazem solicitações de lote.

```
/*
 * 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.
            }
        }
    }
}
```

Monitorar métricas de volume da execução de exemplo

O Amazon SQS gera automaticamente métricas de volume para mensagens enviadas, recebidas e excluídas. Você pode acessar essas métricas e outras por meio da guia Monitoramento da sua fila ou no CloudWatch console.

> **Note**
>
> As métricas podem levar até 15 minutos após a fila começar para ficar disponíveis.

Como trabalhar com o JMS e o Amazon SQS

A biblioteca de mensagens Java do Amazon SQS é uma interface Java Message Service (JMS) para o Amazon SQS que permite aproveitar o Amazon SQS em aplicações que já utilizam JMS. A interface permite que você use o Amazon SQS como o provedor de JMS com o mínimo de alterações no código. Junto com o AWS SDK for Java, a biblioteca de mensagens Java do Amazon SQS permite criar conexões e sessões JMS, bem como produtores e consumidores que enviam e recebem mensagens de e para filas do Amazon SQS.

A biblioteca suporta o envio e o recebimento de mensagens para uma fila (o point-to-point modelo JMS) de acordo com a especificação JMS 1.1. A biblioteca oferece suporte ao envio de mensagens de texto, de byte ou de objeto de forma síncrona para filas do Amazon SQS. A biblioteca também dá suporte ao recebimento de objetos de forma síncrona ou assíncrona.

Para mais informações sobre recursos da biblioteca de mensagens Java do Amazon SQS compatíveis com a especificação JMS 1.1, consulte O Amazon SQS suportou implementações do JMS 1.1 e as Perguntas frequentes do Amazon SQS.

Tópicos

Pré-requisitos para trabalhar com o JMS e o Amazon SQS

Antes de começar, você deve cumprir os seguintes pré-requisitos:

- SDK para Java

  Há duas maneiras de incluir o SDK for Java no seu projeto:

  - Faça download e instale o SDK for Java.
  - Use o Maven para obter a biblioteca de mensagens Java do Amazon SQS.

> **Note**
>
> O SDK for Java é incluído como uma dependência.
> O SDK for Java e a biblioteca cliente Java estendida para o Amazon SQS exigem o J2SE Development Kit 8.0 ou posterior.

Para obter mais informações sobre como fazer download do SDK for Java, consulte SDK for Java.

- Biblioteca de Mensagens Java do Amazon SQS

  Se você não usar o Maven, é necessário adicionar o pacote `amazon-sqs-java-messaging-lib.jar` ao caminho da classe Java. Para obter mais informações sobre como fazer download da biblioteca, consulte Biblioteca de mensagens Java do Amazon SQS.

  > **Note**
  >
  > A biblioteca de mensagens Java do Amazon SQS inclui suporte para Maven e Spring Framework.
  > Para exemplos de código que usam Maven, Spring Framework e a biblioteca de mensagens Java do Amazon SQS, consulte Exemplos de trabalho em Java para usar o JMS com filas padrão do Amazon SQS.
  >
  > ```
  > <dependency>
  >   <groupId>com.amazonaws</groupId>
  >   <artifactId>amazon-sqs-java-messaging-lib</artifactId>
  >   <version>1.0.4</version>
  >   <type>jar</type>
  > </dependency>
  > ```

- Fila do Amazon SQS

  Crie uma fila usando o AWS Management Console para Amazon SQS, a API ou `CreateQueue` o cliente Amazon SQS encapsulado incluído na Biblioteca de Mensagens Java do Amazon SQS.

  - Para obter mais informações sobre como criar uma fila com o Amazon SQS usando o AWS Management Console ou a API `CreateQueue`, consulte Criar uma fila.

- Para obter mais informações sobre o uso da biblioteca de mensagens Java do Amazon SQS, consulte Conceitos básicos da biblioteca de mensagens Java do Amazon SQS.

Conceitos básicos da biblioteca de mensagens Java do Amazon SQS

Para começar a usar o Java Message Service (JMS) com o Amazon SQS, use os exemplos de código nesta seção. As seções a seguir mostram como criar uma conexão e uma sessão JMS, e como enviar e receber uma mensagem.

O objeto do cliente encapsulado do Amazon SQS incluído na biblioteca de mensagens Java do Amazon SQS verifica se uma fila do Amazon SQS existe. Se a fila não existir, o cliente a criará.

Criação de uma conexão JMS

1. Crie uma connection factory e chame o método `createConnection` contra a factory.

```
// 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();
```

   A classe `SQSConnection` estende `javax.jms.Connection`. Junto com os métodos de conexão JMS padrão, `SQSConnection` oferece métodos adicionais, como `getAmazonSQSClient` e `getWrappedAmazonSQSClient`. Os dois métodos permitem que você execute operações administrativas não incluídas na especificação JMS, como a criação de novas filas. Contudo, o método `getWrappedAmazonSQSClient` também fornece uma versão encapsulada do cliente do Amazon SQS usada pela conexão atual. O wrapper transforma cada exceção de um cliente em um `JMSException`, permitindo que ele seja mais facilmente usado pelo código existente que espera ocorrências de `JMSException`.

2. Você pode usar objetos de cliente retornados de `getAmazonSQSClient` e de `getWrappedAmazonSQSClient` para executar operações administrativas não incluídas na especificação do JMS (por exemplo, você pode criar uma fila do Amazon SQS).

   Se você tiver um código que espera exceções JMS, deve usar `getWrappedAmazonSQSClient`:

   - Se você usar `getWrappedAmazonSQSClient`, o objeto de cliente retornado transformará todas as exceções em exceções JMS.
   - Se você usar `getAmazonSQSClient`, todas as exceções serão exceções do Amazon SQS.

Criar uma fila do Amazon SQS

O objeto de cliente encapsulado verifica se uma fila do Amazon SQS existe.

Se a fila não existir, o cliente a criará. Se a fila existir, a função não retornará nada. Para obter mais informações, consulte a seção "Criar uma fila, se necessário" no exemplo TextMessageSender.java.

Como criar uma fila padrão

```
// 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");
}
```

Para criar uma fila 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**
>
> O nome de uma fila FIFO deve terminar com o sufixo `.fifo`.
> Para obter mais informações sobre o atributo `ContentBasedDeduplication`, consulte Processamento de exatamente uma vez no Amazon SQS.

Envio de mensagens de forma síncrona

1. Quando a conexão e a fila do Amazon SQS subjacente estiverem prontas, crie uma sessão JMS sem transação com o modo `AUTO_ACKNOWLEDGE`.

```
// Create the nontransacted session with AUTO_ACKNOWLEDGE mode
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
```

2. Para enviar uma mensagem de texto para a fila, crie uma identidade de fila JMS e um produtor de mensagem.

```
// 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. Crie uma mensagem de texto e envie-a para a fila.

   - Para enviar uma mensagem para uma fila padrão, você não precisa definir parâmetros adicionais.

```
// 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 uma mensagem para uma fila FIFO, você deve definir o ID do grupo de mensagens. Você também pode definir um ID de eliminação de duplicação de mensagem. Para ter mais informações, consulte Termos-chave do 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"));
```

Recebimento de mensagens de forma síncrona

1. Para receber mensagens, crie um consumidor para a mesma fila e invoque o método `start`.

   Você também pode chamar o método `start` na conexão a qualquer momento. No entanto, o consumidor não começa a receber mensagens até você chamá-lo.

```
// Create a consumer for the 'MyQueue'
MessageConsumer consumer = session.createConsumer(queue);
// Start receiving incoming messages
connection.start();
```

2. Chame o método `receive` no consumidor com um tempo limite definido como 1 segundo e imprima o conteúdo da mensagem recebida.

   - Após receber uma mensagem de uma fila padrão, você pode acessar o conteúdo da mensagem.

```
// 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());
}
```

- Após receber uma mensagem de uma fila FIFO, você pode acessar o conteúdo da mensagem e outros atributos de mensagem específicos à FIFO, como o ID do grupo de mensagens, o ID de eliminação de duplicação de mensagens e o número de sequência. Para ter mais informações, consulte Termos-chave do 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. Feche a conexão e a sessão.

```
// Close the connection (and the session).
connection.close();
```

A saída será semelhante à seguinte:

```
JMS Message ID:8example-588b-44e5-bbcf-d816example2
Received: Hello World!
```

> **Note**
>
> Você pode usar o Spring Framework para inicializar esses objetos.

Para obter mais informações, consulte `SpringExampleConfiguration.xml`, `SpringExample.java` e as outras classes auxiliares em `ExampleConfiguration.java` e `ExampleCommon.java` na seção Exemplos de trabalho em Java para usar o JMS com filas padrão do Amazon SQS.

Para exemplos completos de envio e recebimento de objetos, consulte TextMessageSender.java e SyncMessageReceiver.java.

Recebimento de mensagens de forma assíncrona

No exemplo em Conceitos básicos da biblioteca de mensagens Java do Amazon SQS, uma mensagem é enviada para `MyQueue` e recebida de forma síncrona.

O exemplo a seguir mostra como receber as mensagens de forma assíncrona por meio de um listener.

1. Implemente a interface `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();
        }
    }
}
```

   O método `onMessage` da interface `MessageListener` é chamado quando você recebe uma mensagem. Nesta implementação de listener, o texto armazenado na mensagem é impresso.

2. Em vez de explicitamente chamar o método `receive` no consumidor, defina o listener da mensagem do consumidor como uma instância da implementação `MyListener`. O thread principal aguarda um 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);
```

As demais etapas são idênticas às do exemplo Conceitos básicos da biblioteca de mensagens Java do Amazon SQS. Para um exemplo completo de um consumidor assíncrono, consulte `AsyncMessageReceiver.java` em Exemplos de trabalho em Java para usar o JMS com filas padrão do Amazon SQS.

A saída deste exemplo é similar ao seguinte:

```
JMS Message ID:8example-588b-44e5-bbcf-d816example2
Received: Hello World!
```

Uso do modo de reconhecimento do cliente

O exemplo em Conceitos básicos da biblioteca de mensagens Java do Amazon SQS usa o modo `AUTO_ACKNOWLEDGE` em que cada mensagem recebida é confirmada automaticamente (e, portanto, excluída da fila do Amazon SQS subjacente).

1. Para explicitamente reconhecer as mensagens depois de processadas, você deve criar a sessão com o modo `CLIENT_ACKNOWLEDGE`.

   ```
   // Create the non-transacted session with CLIENT_ACKNOWLEDGE mode.
   Session session = connection.createSession(false, Session.CLIENT_ACKNOWLEDGE);
   ```

2. Quando a mensagem é recebida, exiba-a e confirme-a explicitamente.

   ```
   // 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**
>
> Nesse modo, quando uma mensagem é confirmada, todas as mensagens recebidas antes desta mensagem são implicitamente confirmadas. Por exemplo, se 10 mensagens são recebidas e apenas a 10ª mensagem é reconhecida (na ordem em que as mensagens são recebidas), todas as nove mensagens anteriores também são reconhecidas.

As demais etapas são idênticas às do exemplo Conceitos básicos da biblioteca de mensagens Java do Amazon SQS. Para um exemplo completo de um consumidor síncrono com modo de reconhecimento do cliente, consulte `SyncMessageReceiverClientAcknowledge.java` em Exemplos de trabalho em Java para usar o JMS com filas padrão do Amazon SQS.

A saída deste exemplo é similar ao seguinte:

```
JMS Message ID:4example-aa0e-403f-b6df-5e02example5
Received: Hello World!
Acknowledged: ID:4example-aa0e-403f-b6df-5e02example5
```

Uso do modo de reconhecimento não ordenado

Ao usar o modo `CLIENT_ACKNOWLEDGE`, todas as mensagens recebidas antes de uma mensagem explicitamente reconhecida são automaticamente reconhecidas. Para ter mais informações, consulte Uso do modo de reconhecimento do cliente.

A biblioteca de mensagens Java do Amazon SQS fornece outro modo de confirmação. Ao usar o modo `UNORDERED_ACKNOWLEDGE`, todas as mensagens recebidas devem ser individual e explicitamente reconhecidas pelo cliente, independentemente de sua ordem de recebimento. Para fazer isso, cria uma sessão com o modo `UNORDERED_ACKNOWLEDGE`.

```
// Create the non-transacted session with UNORDERED_ACKNOWLEDGE mode.
Session session = connection.createSession(false, SQSSession.UNORDERED_ACKNOWLEDGE);
```

As etapas restantes são idênticas às do exemplo Uso do modo de reconhecimento do cliente. Para um exemplo completo de um consumidor síncrono com o modo `UNORDERED_ACKNOWLEDGE`, consulte `SyncMessageReceiverUnorderedAcknowledge.java`.

Neste exemplo, a saída é similar ao seguinte:

```
JMS Message ID:dexample-73ad-4adb-bc6c-4357example7
Received: Hello World!
Acknowledged: ID:dexample-73ad-4adb-bc6c-4357example7
```

Usando o Java Message Service com outros clientes do Amazon SQS

O uso do cliente Amazon SQS Java Message Service (JMS) com o AWS SDK limita o tamanho da mensagem do Amazon SQS a 256 KB. No entanto, você pode criar um provedor JMS usando qualquer cliente do Amazon SQS. Por exemplo, você pode usar o cliente JMS com a biblioteca cliente Java estendida para o Amazon SQS para enviar uma mensagem do Amazon SQS que contenha uma referência à carga útil da mensagem (até 2 GB) no Amazon S3. Para ter mais informações, consulte Gerenciando grandes mensagens do Amazon SQS usando Java e Amazon S3.

O seguinte exemplo de código Java cria o provedor JMS para a biblioteca cliente estendida:

```
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);
```

O exemplo de código Java a seguir cria a connection factory:

```
// 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();
```

Exemplos de trabalho em Java para usar o JMS com filas padrão do Amazon SQS

Veja a seguir exemplos de código que mostram como usar o JMS (Java Message Service) com filas padrão do Amazon SQS. Para obter mais informações sobre como trabalhar com filas FIFO, consulte Para criar uma fila FIFO, Envio de mensagens de forma síncrona e Recebimento de mensagens de forma síncrona. (O recebimento de mensagens de forma síncrona é igual para filas padrão e FIFO. No entanto, as mensagens em filas FIFO contêm mais atributos).

ExampleConfiguration.java

O exemplo de código Java SDK v 1.x a seguir define o nome da fila padrão, a região e as credenciais a serem usadas com os outros exemplos 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

O seguinte exemplo de código Java cria um produtor de mensagem 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

O seguinte exemplo de código Java cria um consumidor de mensagem síncrona.

```
/*
 * 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

O seguinte exemplo de código Java cria um consumidor de mensagem assíncrona.

```
/*
 * 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

O seguinte exemplo de código Java cria um consumidor síncrono com modo de reconhecimento do 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

O seguinte exemplo de código Java cria um consumidor síncrono com modo de reconhecimento não ordenado.

```
/*
```

```
 * 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

O seguinte exemplo de código XML é um arquivo de configuração 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

O seguinte exemplo de código Java usa o arquivo de configuração bean para inicializar seus 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

O código de exemplo Java a seguir verifica se existe uma fila do Amazon SQS e, se não existir, cria uma. Ele também inclui código de registro de exemplo.

```
/*
 * 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() );
        }
    }
}
```

O Amazon SQS suportou implementações do JMS 1.1

A biblioteca de mensagens Java do Amazon SQS oferece suporte às seguintes implantações do JMS 1.1. Para mais informações sobre os recursos compatíveis e capacidade da biblioteca de mensagens Java do Amazon SQS, consulte as Perguntas frequentes do Amazon SQS.

Interfaces comuns com suporte

- `Connection`
- `ConnectionFactory`
- `Destination`
- `Session`
- `MessageConsumer`
- `MessageProducer`

Tipos de mensagens com suporte

- `ByteMessage`
- `ObjectMessage`
- `TextMessage`

Modos de reconhecimento de mensagens com suporte

- `AUTO_ACKNOWLEDGE`
- `CLIENT_ACKNOWLEDGE`
- `DUPS_OK_ACKNOWLEDGE`
- `UNORDERED_ACKNOWLEDGE`

> **Note**
>
> O modo `UNORDERED_ACKNOWLEDGE` não faz parte da especificação JMS 1.1. Esse modo ajuda o Amazon SQS a permitir que um cliente JMS reconheça explicitamente uma mensagem.

Cabeçalhos definidos pelo JMS e propriedades reservadas

Para enviar mensagens

Ao enviar mensagens, você pode definir os seguintes cabeçalhos e propriedades para cada mensagem:

- `JMSXGroupID` (obrigatório para filas FIFO, não permitido para filas padrão)
- `JMS_SQS_DeduplicationId` (opcional para filas FIFO, não permitido para filas padrão)

Quando você envia mensagens, o Amazon SQS define os seguintes cabeçalhos e propriedades para cada uma:

- `JMSMessageID`
- `JMS_SQS_SequenceNumber` (somente para filas FIFO)

Para receber mensagens

Quando você recebe mensagens, o Amazon SQS define os seguintes cabeçalhos e propriedades para cada uma:

- `JMSDestination`
- `JMSMessageID`
- `JMSRedelivered`
- `JMSXDeliveryCount`
- `JMSXGroupID` (somente para filas FIFO)
- `JMS_SQS_DeduplicationId` (somente para filas FIFO)
- `JMS_SQS_SequenceNumber` (somente para filas FIFO)

Tutoriais do Amazon SQS

Esta seção fornece tutoriais que você pode usar para explorar os recursos e funcionalidades do Amazon SQS.

Tópicos

Criando uma fila do Amazon SQS usando AWS CloudFormation

Você pode usar o AWS CloudFormation console e um modelo JSON (ou YAML) para criar uma fila do Amazon SQS. Para obter mais informações, consulte Trabalhar com modelos do AWS CloudFormation e o recurso `AWS::SQS::Queue` no Guia do usuário do AWS CloudFormation .

Para usar AWS CloudFormation para criar uma fila do Amazon SQS.

1. Copie o seguinte código JSON em um arquivo denominado `MyQueue.json`. Para criar uma fila padrão, omita as propriedades `FifoQueue` e `ContentBasedDeduplication`. Para obter mais informações sobre a eliminação de duplicação baseada em conteúdo, consulte Processamento de exatamente uma vez no Amazon SQS.

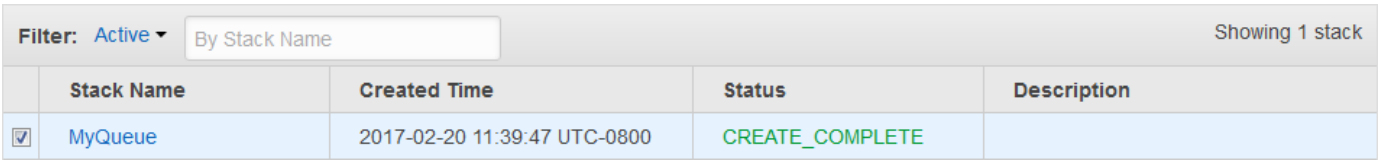
   > **Note**
   >
   > O nome de uma fila FIFO deve terminar com o sufixo `.fifo`.

```
{
   "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. Faça login no console do AWS CloudFormation e, em seguida, selecione Create Stack (Criar pilha).
3. No painel Specify Template (Especificar modelo), escolha Upload a template file (Fazer upload de um arquivo de modelo), selecione o arquivo `MyQueue.json` e escolha Next (Próximo).
4. Na página Specify Details, digite `MyQueue` em Stack Name e escolha Next.
5. Na página Options (Opções), escolha Next (Avançar).
6. Na página Review (Revisar), escolha Create (Criar).

   AWS CloudFormation começa a criar a `MyQueue` pilha e exibe o status CREATE_IN_PROGRESS. Quando o processo é concluído, o AWS CloudFormation exibe o status CREATE_COMPLETE.



7. (Opcional) Para exibir o nome, a URL e o nome de recurso da Amazon (ARN) da fila, escolha o nome da pilha e, em seguida, na próxima página, expanda a seção Outputs.

Tutorial: Envio de uma mensagem a uma fila do Amazon SQS pela Amazon Virtual Private Cloud

Neste tutorial, você aprenderá como enviar mensagens para uma fila do Amazon SQS por uma rede privada e segura. Essa rede consiste em uma VPC que contém uma instância do Amazon EC2. A instância se conecta ao Amazon SQS por meio de uma interface de endpoint da VPC, permitindo que você se conecte à instância do Amazon EC2 e envie mensagens para a fila do Amazon SQS mesmo que a rede esteja desconectada da Internet pública. Para ter mais informações, consulte Endpoints da Amazon Virtual Private Cloud para o Amazon SQS.

Important

- Você pode usar a Amazon Virtual Private Cloud somente com endpoints HTTPS do Amazon SQS.
- Ao configurar o Amazon SQS para enviar mensagens pela Amazon VPC, habilite o DNS privado e especifique endpoints no formato `sqs.us-east-2.amazonaws.com`.
- O DNS privado não oferece suporte a endpoints legados, como `queue.amazonaws.com` ou `us-east-2.queue.amazonaws.com`.

Tópicos

Etapa 1: criar um par de chaves do Amazon EC2

Um par de chaves permite que você se conecte a uma instância do Amazon EC2. Esse par consiste em uma chave pública que criptografa suas informações de login e uma chave privada que as descriptografa.

1. Faça login no console do Amazon EC2.
2. No menu de navegação, em Network & Security (Rede e segurança), selecione Key Pairs (Pares de chaves).
3. Escolha Criar par de chaves.
4. Na caixa de diálogo Create Key Pair (Criar par de chaves), para Key pair name (Nome do par de chaves), insira `SQS-VPCE-Tutorial-Key-Pair` e selecione Create (Criar).
5. O navegador faz download do arquivo da chave privada `SQS-VPCE-Tutorial-Key-Pair.pem` automaticamente.

   > **Important**
   >
   > Salve esse arquivo em um lugar seguro. O EC2 não gera um segundo arquivo `.pem` para o mesmo par de chaves.

6. Para permitir que um cliente SSH se conecte à sua instância do EC2, defina as permissões do arquivo da chave privada para que somente o seu usuário possa ter permissões de leitura para ele, por exemplo:

   ```
   chmod 400 SQS-VPCE-Tutorial-Key-Pair.pem
   ```

Etapa 2: criar AWS recursos

Para configurar a infraestrutura necessária, você deve usar um AWS CloudFormation modelo, que é um modelo para criar uma pilha composta por AWS recursos, como instâncias do Amazon EC2 e filas do Amazon SQS.

A pilha deste tutorial inclui os seguintes recursos:

- Uma VPC e os recursos de rede associados, incluindo uma sub-rede, um grupo de segurança, um gateway da Internet e uma tabela de rotas.

- Uma instância do Amazon EC2 executada na sub-rede da VPC
- Uma fila do Amazon SQS

1. Baixe o AWS CloudFormation modelo com o nome `SQS-VPCE-Tutorial-CloudFormation.yaml`deGitHub.
2. Faça login no console do AWS CloudFormation.
3. Escolha Create Stack (Criar pilha).
4. Na página Select Template (Selecionar modelo), selecione Upload a template to Amazon S3 (Fazer upload de um modelo no Amazon S3), selecione o arquivo `SQS-VPCE-SQS-Tutorial-CloudFormation.yaml` e, então, selecione Next (Próximo).
5. Na página Specify Details (Especificar detalhes), faça o seguinte:
   a. Para Stack name (Nome da pilha), insira `SQS-VPCE-Tutorial-Stack`.
   b. Para KeyName, escolha SQS-VPCE-Tutorial-Key-pair.
   c. Escolha Próximo.
6. Na página Options (Opções), escolha Next (Avançar).
7. Na página de revisão, na seção Capacidades, escolha Eu reconheço que AWS CloudFormation pode criar recursos do IAM com nomes personalizados. e, em seguida, escolha Criar.

AWS CloudFormation começa a criar a pilha e exibe o status CREATE_IN_PROGRESS. Quando o processo é concluído, o AWS CloudFormation exibe o status CREATE_COMPLETE.

Etapa 3: confirmar que sua instância do EC2 não é acessível publicamente

Seu AWS CloudFormation modelo inicia uma instância do EC2 nomeada `SQS-VPCE-Tutorial-EC2-Instance` em sua VPC. Essa instância do EC2 não permite tráfego de saída e não é capaz de enviar mensagens para o Amazon SQS. Para verificar isso, você deve se conectar à instância, tentar se conectar a um endpoint público e, então, tentar enviar uma mensagem para o Amazon SQS.

1. Faça login no console do Amazon EC2.
2. No menu de navegação, em Instances (Instâncias), selecione Instances (Instâncias).
3. Selecione SQS-VPCE-. Tutorial-EC2Instance
4. Copie o nome de host em Public DNS (IPv4) (DNS público (IPv4)), por exemplo, ec2-203-0-113-0.us-west-2.compute.amazonaws.com.

5. A partir do diretório que contém o par de chaves criado anteriormente, conecte-se à instância usando o comando a seguir, por exemplo:

```
ssh -i SQS-VPCE-Tutorial-Key-Pair.pem ec2-user@ec2-203-0-113-0.us-east-2.compute.amazonaws.com
```

6. Tente conectar-se a qualquer endpoint público, por exemplo:

```
ping amazon.com
```

   A tentativa de conexão falha, conforme esperado.

7. Faça login no console do Amazon SQS.
8. Na lista de filas, selecione a fila criada pelo seu AWS CloudFormation modelo, por exemplo, VPCE-SQS-Tutorial-Stack-CFQueue-1abcdefgh2ijk.
9. Na tabela Details (Detalhes), copie o URL, por exemplo, https://sqs.us-east-2.amazonaws.com/123456789012/.
10. A partir de sua instância do EC2, tente publicar uma mensagem na fila usando o comando a seguir, por exemplo:

```
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."
```

    A tentativa de envio falha, conforme esperado.

> **Important**
>
> Posteriormente, ao criar um endpoint da VPC para o Amazon SQS, sua tentativa de envio será bem-sucedida.

Etapa 4: criar um endpoint da Amazon VPC para o Amazon SQS

Para conectar sua VPC ao Amazon SQS, você precisa definir uma interface de endpoint da VPC. Depois de adicionar o endpoint, você poderá usar a API do Amazon SQS a partir da instância do EC2 em sua VPC. Isso permite que você envie mensagens para uma fila na AWS rede sem cruzar a Internet pública.

> **Note**
>
> A instância EC2 ainda não tem acesso a outros AWS serviços e endpoints na Internet.

1. Faça login no console da Amazon VPC.
2. No menu de navegação, selecione Endpoints.
3. Escolha Criar Endpoint.
4. Na página Create Endpoint (Criar endpoint), em Service Name (Nome do serviço), selecione o nome do serviço para o Amazon SQS.

   > **Note**
   >
   > Os nomes dos serviços variam de acordo com a AWS região atual. Por exemplo, se você estiver no Leste dos EUA (Ohio), o nome do serviço será com.amazonaws.***us-east-2***.sqs.

5. Para VPC, selecione SQS-VPCE-Tutorial-VPC.
6. Para Subnets (Sub-redes), selecione a sub-rede cujo Subnet ID (ID da sub-rede) contenha SQS-VPCE-Tutorial-Subnet.
7. Para Security group (Grupo de segurança), selecione Select security groups (Selecionar grupos de segurança) e selecione o grupo de segurança cujo Group Name (Nome do grupo) contenha SQS VPCE Tutorial Security Group.
8. Escolha Criar endpoint.

   O VPC endpoint de interface é criado e o ID dele é exibido, por exemplo, vpce-0ab1cdef2ghi3j456k.
9. Escolha Fechar.

   O console da Amazon VPC abre a página Endpoints.

A Amazon VPC começa a criar o endpoint e exibe o status pending (pendente). Quando o processo é concluído, a Amazon VPC exibe o status available (disponível).

Etapa 5: enviar uma mensagem para sua fila do Amazon SQS

Agora que a VPC inclui um endpoint para o Amazon SQS, você pode se conectar à instância do EC2 e enviar mensagens para a fila.

1. Reconecte-se à instância do EC2, por exemplo:

```
ssh -i SQS-VPCE-Tutorial-Key-Pair.pem ec2-user@ec2-203-0-113-0.us-
east-2.compute.amazonaws.com
```

2. Tente publicar uma mensagem na fila novamente usando o comando a seguir, por exemplo:

```
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."
```

   A tentativa de envio é bem-sucedida e o resumo MD5 do corpo da mensagem e o ID da mensagem são exibidos, por exemplo:

```
{
 "MD5OfMessageBody": "a1bcd2ef3g45hi678j90klmn12p34qr5",
 "MessageId": "12345a67-8901-2345-bc67-d890123e45fg"
}
```

Para obter informações sobre como receber e excluir a mensagem da fila criada pelo seu AWS CloudFormation modelo (por exemplo, VPCE-SQS-Tutorial-Stack-CFQueue-1abcdefgh2ijk), consulte. Recebendo e excluindo uma mensagem no Amazon SQS

Para obter informações sobre como excluir seus recursos, consulte o seguinte:

- Excluir um endpoint da VPC no Guia do usuário da Amazon VPC
- Excluir uma fila do Amazon SQS
- Encerre sua instância no Guia do usuário do Amazon EC2
- Excluir a VPC no Guia do usuário da Amazon VPC
- Excluindo uma pilha no AWS CloudFormation console no Guia do usuário AWS CloudFormation
- Excluindo seu par de chaves no Guia do usuário do Amazon EC2

Solução de problemas no Amazon SQS

Os tópicos a seguir fornecem conselhos sobre solução de problemas e erros comuns que você pode encontrar ao usar o console do Amazon SQS, a API do Amazon SQS ou outras ferramentas com o Amazon SQS. Se encontrar um problema que não esteja listado aqui, você poderá usar o botão Feedback desta página para relatá-lo.

Para obter mais orientações sobre solução de problemas e respostas a perguntas comuns de suporte, acesse a Central de Conhecimento da AWS.

Tópicos

Solucionar um erro de acesso negado no Amazon SQS

Os tópicos a seguir abordam as causas `AccessDenied` ou `AccessDeniedException` erros mais comuns nas chamadas de API do Amazon SQS. Para obter mais informações sobre como solucionar esses erros, consulte Como soluciono erros de "" ou AccessDenied "AccessDeniedExceção" em chamadas de API do Amazon SQS? no Guia do Centro de AWS Conhecimento.

Exemplos de mensagens de erro:

```
An error occurred (AccessDenied) when calling the SendMessage operation: Access to
        the resource https://sqs.us-east-1.amazonaws.com/ is denied.
```

- ou -

```
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.
```

Tópicos

- Política de filas do Amazon SQS e política do IAM
- AWS Key Management Service permissões
- Política de endpoint da VPC
- Política de controle de serviços da organização

Política de filas do Amazon SQS e política do IAM

Para verificar se o solicitante tem as permissões adequadas para realizar uma operação do Amazon SQS, faça o seguinte:

- Identifique o diretor do IAM que está fazendo a chamada de API do Amazon SQS. Se o diretor do IAM for da mesma conta, a política de filas do Amazon SQS ou a política de AWS Identity and Access Management (IAM) devem incluir permissões para permitir explicitamente o acesso à ação.
- Se o principal for uma entidade do IAM:
  - Você pode identificar seu usuário ou função do IAM verificando o canto superior direito do AWS Management Console ou usando o `aws sts get-caller-identity`comando.
  - Verifique as políticas do IAM relacionadas ao perfil ou usuário do IAM. É possível usar um dos seguintes métodos:
    - Teste as políticas do IAM com o IAM Policy Simulator.
    - Analise os diferentes tipos de política do IAM.
  - Se necessário, edite a política de usuário do IAM.
  - Verifique a política de filas e edite, se necessário.
- Se o principal for um AWS serviço, a política de filas do Amazon SQS deve permitir explicitamente o acesso.
- Se o principal for um principal entre contas, tanto a política de filas do Amazon SQS quanto a política do IAM devem permitir explicitamente o acesso.
- Se a política usar um elemento condicional, verifique se a condição restringe o acesso.

> **Important**
>
> Uma negação explícita em qualquer política substitui uma permissão explícita. Aqui estão alguns exemplos básicos das políticas do Amazon SQS.

AWS Key Management Service permissões

Se sua fila do Amazon SQS tiver a criptografia do lado do servidor (SSE) ativada com um cliente gerenciado AWS KMS key, as permissões deverão ser concedidas tanto aos produtores quanto aos consumidores. Para confirmar se uma fila está criptografada, você pode usar o **`KmsMasterKeyId`** atributo da **`GetQueueAttributes`**API ou do console da fila em Criptografia.

- Permissões necessárias para produtores:

```
{
"Effect": "Allow",
"Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
],
"Resource": "<Key ARN>"
}
```

- Permissões necessárias para consumidores:

```
{
"Effect": "Allow",
"Action": [
    "kms:Decrypt"
],
"Resource": "<Key ARN>"
}
```

- Permissões necessárias para acesso entre contas:

```
{
"Effect": "Allow",
"Action": [
    "kms:DescribeKey",
    "kms:Decrypt",
```

```
    "kms:ReEncrypt",
    "kms:GenerateDataKey"
],
"Resource": "<Key ARN>"
}
```

Você pode usar qualquer um dos seguintes para habilitar a criptografia para uma fila do Amazon SQS:

- SSE-Amazon SQS (chave de criptografia criada e gerenciada pelo serviço Amazon SQS.)
- AWS chave padrão gerenciada (alias/aws/sqs)
- Chave gerenciada pelo cliente

No entanto, se você estiver usando uma chave KMS AWS gerenciada, não poderá modificar a política de chaves padrão. Portanto, para fornecer acesso a outros serviços e contas cruzadas, use a chave gerenciada pelo cliente. Isso permite que você edite a política de chaves.

Política de endpoint da VPC

Se você acessar o Amazon SQS por meio de um endpoint Amazon Virtual Private Cloud (Amazon VPC), a política de endpoint VPC do Amazon SQS deve permitir o acesso. Você pode criar uma política para endpoints Amazon VPC para Amazon SQS, onde você pode especificar o seguinte:

1. A entidade principal que pode executar ações.
2. As ações que podem ser executadas.
3. Os recursos sobre os quais as ações podem ser realizadas.

No exemplo a seguir, a política de endpoint da VPC especifica que o usuário *MyUser*do IAM tem permissão para enviar mensagens para a fila do Amazon SQS. *MyQueue* Outras ações, usuários do IAM e recursos do Amazon SQS têm acesso negado por meio do VPC endpoint.

```
{
   "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 controle de serviços da organização

Se você Conta da AWS pertence a uma organização, AWS Organizations as políticas podem impedir que você acesse suas filas do Amazon SQS. Por padrão, AWS Organizations as políticas não bloqueiam nenhuma solicitação para o Amazon SQS. No entanto, certifique-se de que suas AWS Organizations políticas não tenham sido configuradas para bloquear o acesso às filas do Amazon SQS. Para obter instruções sobre como verificar suas AWS Organizations políticas, consulte Listar todas as políticas no Guia AWS Organizations do usuário.

Solucionar problemas de erros da API do Amazon SQS

Os tópicos a seguir abordam os erros mais comuns retornados ao fazer chamadas de API do Amazon SQS e como solucioná-los.

Tópicos

- QueueDoesNotExist erro
- InvalidAttributeValue erro
- ReceiptHandle erro

QueueDoesNotExist erro

Esse erro será retornado quando o serviço Amazon SQS não conseguir encontrar a fila mencionada para a ação do Amazon SQS.

Possíveis causas e mitigações:

- Região incorreta: revise a configuração do cliente Amazon SQS para confirmar se você configurou a região correta no cliente. Quando você não configura uma região no cliente, o SDK ou AWS CLI escolhe a região no arquivo de configuração ou na variável de ambiente. Se o SDK não encontrar uma região no arquivo de configuração, ele definirá a região como us-east-1 por padrão.
- A fila pode ter sido excluída recentemente: se a fila foi excluída antes da chamada da API ser feita, a chamada da API retornará esse erro. Verifique se CloudTrail há alguma DeleteQueueoperação antes do momento do erro.

- Problemas de permissão: se o usuário ou a função solicitante AWS Identity and Access Management (IAM) não tiver as permissões necessárias, você poderá receber o seguinte erro:

  ```
  The specified queue does not exist or you do not have access to it.
  ```

  Verifique as permissões e faça a chamada de API com as permissões corretas.

Para obter mais detalhes sobre a solução do `QueueDoesNotExist` erro, consulte Como soluciono o QueueDoesNotExist erro ao fazer chamadas de API para minha fila do Amazon SQS? no Guia do Centro de AWS Conhecimento.

InvalidAttributeValue erro

Esse erro será retornado ao atualizar a política de recursos de fila do Amazon SQS ou propriedades com uma política ou um principal incorreto.

Possíveis causas e mitigações:

- Política de recursos inválida: verifique se a política de recursos tem todos os campos obrigatórios. Para obter mais informações, consulte Referência de elementos de política JSON do IAM e Validação de políticas do IAM. Você também pode usar o gerador de políticas do IAM para criar e testar uma política de recursos do Amazon SQS. Certifique-se de que a política esteja no formato JSON.
- Principal inválido: verifique se o `Principal` elemento existe na política de recursos e se o valor é válido. Se o `Principal` elemento de política de recursos do Amazon SQS incluir uma entidade do IAM, certifique-se de que a entidade exista antes de usar a política. O Amazon SQS valida a política de recursos e verifica a entidade do IAM. Se a entidade do IAM não existir, você receberá um erro. Para confirmar as entidades do IAM, use as `GetUser`APIs `GetRole`e.

Para obter informações adicionais sobre como solucionar um `InvalidAttributeValue` erro, consulte Como soluciono o QueueDoesNotExist erro ao fazer chamadas de API para minha fila do Amazon SQS? no Guia do Centro de AWS Conhecimento.

ReceiptHandle erro

Ao fazer uma chamada `DeleteMessage`à API, o erro `ReceiptHandleIsInvalid` `InvalidParameterValue` pode ser retornado se o identificador do recibo estiver incorreto ou expirado.

- ReceiptHandleIsInvalid erro: se o identificador do recibo estiver incorreto, você receberá um erro semelhante a este exemplo:

```
An error occurred (ReceiptHandleIsInvalid) when calling the DeleteMessage operation:
 The input receipt handle <YOUR RECEIPT HANDLE> is not a valid receipt handle.
```

- InvalidParameterValue erro: se o identificador do recibo expirar, você receberá um erro semelhante a este exemplo:

```
An error occurred (InvalidParameterValue) when calling the DeleteMessage operation:
 Value <YOUR RECEIPT HANDLE> for parameter ReceiptHandle is invalid. Reason: The
 receipt handle has expired.
```

Possíveis causas e mitigações:

O identificador do recibo é criado para cada mensagem recebida e só é válido para o período de tempo limite de visibilidade. Quando o período de tempo limite de visibilidade expira, a mensagem fica visível na fila para os consumidores. Ao receber a mensagem novamente do consumidor, você recebe um novo identificador de recibo. Para evitar erros incorretos ou expirados no identificador de recibo, use o identificador de recibo correto para excluir a mensagem dentro do período de tempo limite de visibilidade da fila do Amazon SQS.

Para obter informações adicionais sobre como solucionar um `ReceiptHandle` erro, consulte Como soluciono erros "" e ReceiptHandle IsInvalid "InvalidParameterValue” ao usar a chamada de API do Amazon DeleteMessage SQS? no Guia do Centro de AWS Conhecimento.

Solucionar problemas de fila de cartas mortas do Amazon SQS e redrive de DLQ

Os tópicos a seguir abordam as causas mais comuns dos problemas de DLQ e DLQ redrive do Amazon SQS e como solucioná-los. Para obter mais informações, consulte Como soluciono problemas de redrive do Amazon SQS DLQ? no Guia do Centro de AWS Conhecimento.

Tópicos

- Problemas de DLQ
- Problemas com o DLQ-Redrive

Problemas de DLQ

Saiba mais sobre problemas comuns de DLQ e como resolvê-los.

Tópicos

Visualizar mensagens usando o console pode fazer com que elas sejam movidas para uma fila de mensagens mortas

O Amazon SQS considera a visualização de uma mensagem no console para a política de redirecionamento da fila correspondente. Portanto, se você visualizar uma mensagem no console o número de vezes especificado na política de redirecionamento da fila correspondente, a mensagem será movida para a fila de mensagens mortas da fila correspondente.

Para ajustar esse comportamento, você pode executar uma das seguintes ações:

- Aumentar a definição de Maximum Receives da política de redirecionamento da fila correspondente.
- Evitar a visualização de mensagens da fila correspondente no console.

O `NumberOfMessagesSent` e o `NumberOfMessagesReceived` de uma fila de mensagens mortas não correspondem

Se você enviar uma mensagem para uma dead letter queue manualmente, ela será capturada pela métrica `NumberOfMessagesSent`. No entanto, se uma mensagem for enviada para uma dead-letter queue como resultado de uma falha na tentativa de processamento, ela não será capturada por essa métrica. Portanto, é possível que os valores de `NumberOfMessagesSent` e `NumberOfMessagesReceived`sejam diferentes.

Criando e configurando um redrive de fila de mensagens mortas

O redirecionamento da fila de mensagens mortas exige que você defina as permissões apropriadas para que o Amazon SQS receba mensagens da fila de mensagens mortas e envie mensagens para a fila de destino. Se você não tiver as permissões corretas, a tarefa de redirecionamento da fila de cartas mortas pode falhar. Você pode ver o status da sua tarefa de redirecionamento de mensagens para corrigir os problemas e tentar novamente.

Tratamento de falhas de mensagens de fila padrão e FIFO

As filas padrão continuam processando as mensagens até o vencimento do período de retenção. Esse processamento contínuo minimiza as chances de a fila ser bloqueada por mensagens não consumidas. Ter um grande número de mensagens que o consumidor deixa de excluir repetidamente pode aumentar os custos e sobrecarregar o hardware. Para manter os custos baixos, mova as mensagens com falha para a fila de mensagens mortas.

As filas padrão também permitem um grande número de mensagens durante o voo. Se a maioria das suas mensagens não puder ser consumida e não for enviada para uma fila de mensagens mortas, sua taxa de processamento de mensagens poderá diminuir. Para manter a eficiência da fila, certifique-se de que seu aplicativo gerencie corretamente o processamento de mensagens.

As filas FIFO garantem o processamento exatamente uma vez, consumindo mensagens de um grupo de mensagens em sequência. Portanto, embora o consumidor possa continuar recuperando mensagens solicitadas de outro grupo de mensagens, o primeiro grupo de mensagens permanece indisponível até que a mensagem que bloqueia a fila seja processada com êxito ou movida para uma fila de mensagens mortas.

Além disso, as filas FIFO permitem um número menor de mensagens em trânsito. Para evitar que sua fila FIFO seja bloqueada por uma mensagem, certifique-se de que seu aplicativo gerencie corretamente o processamento de mensagens.

Para obter mais informações, consulte Cotas de mensagens do Amazon SQS e Trabalhar com mensagens do Amazon SQS.

Problemas com o DLQ-Redrive

Saiba mais sobre problemas comuns do DLQ-Redrive e como resolvê-los.

Tópicos

- AccessDenied problema de permissão
- NonExistentQueue erro
- CouldNotDetermineMessageErro na fonte

AccessDenied problema de permissão

O `AccessDenied` erro ocorre quando o redrive do DLQ falha porque a entidade AWS Identity and Access Management (IAM) não tem as permissões necessárias.

Exemplo de mensagem de erro:

```
Failed to create redrive task. Error code: AccessDenied - Queue Permissions to Redrive.
```

As seguintes permissões de API são necessárias para fazer solicitações de recondução de DLQ:

Para iniciar o redirecionamento de uma mensagem:

- Permissões de fila de cartas mortas:
  - `sqs:StartMessageMoveTask`
  - `sqs:ReceiveMessage`
  - `sqs:DeleteMessage`
  - `sqs:GetQueueAttributes`
  - `kms:Decrypt`— Quando a fila de mensagens mortas ou a fila de origem original são criptografadas.
- Permissões da fila de destino:
  - `sqs:SendMessage`
  - `kms:GenerateDataKey`— Quando a fila de destino é criptografada.
  - `kms:Decrypt` — Quando a fila de destino é criptografada.

Para cancelar o redirecionamento de uma mensagem em andamento:

- Permissões de fila de cartas mortas:
  - `sqs:CancelMessageMoveTask`
  - `sqs:ReceiveMessage`
  - `sqs:DeleteMessage`

- `sqs:GetQueueAttributes`
- `kms:Decrypt`— Quando a fila de mensagens mortas ou a fila de origem original são criptografadas.

Para mostrar o status de movimentação de uma mensagem:

- Permissões de fila de cartas mortas:
  - `sqs:ListMessageMoveTasks`
  - `sqs:GetQueueAttributes`

NonExistentQueue erro

O `NonExistentQueue` erro ocorre quando a fila de origem do Amazon SQS não existe ou foi excluída. Verifique e redirecione para uma fila do Amazon SQS que esteja presente.

Exemplo de mensagem de erro:

```
Failed: AWS.SimpleQueueService.NonExistentQueue
```

CouldNotDetermineMessageErro na fonte

O `CouldNotDetermineMessageSource` erro ocorre quando você tenta iniciar um redrive de DLQ com os seguintes cenários:

- Uma mensagem do Amazon SQS enviada diretamente para o DLQ com a API. `SendMessage`
- Uma mensagem do tópico ou função do Amazon Simple Notification Service (Amazon SNS) com o AWS Lambda DLQ configurado.

Para resolver esse erro, escolha Redirecionar para um destino personalizado ao iniciar o redrive. Em seguida, insira o ARN da fila do Amazon SQS para mover todas as mensagens do DLQ para a fila de destino.

Exemplo de mensagem de erro:

```
Failed: CouldNotDetermineMessageSource
```

Solucionar problemas de limitação de FIFO no Amazon SQS

Por padrão, as filas FIFO suportam 300 transações por segundo, por ação de API para `SendMessageReceiveMessage`, e. `DeleteMessage` Solicitações acima de 300 TPS recebem o `ThrottlingException` erro mesmo se as mensagens na fila estiverem disponíveis. Para mitigar isso, você pode usar os seguintes métodos:

- Habilite alta taxa de transferência para filas FIFO no Amazon SQS.
- Use as ações em lote da API Amazon SQS `SendMessageBatchDeleteMessageBatch`, e `ChangeMessageVisibilityBatch` para aumentar o limite de TPS de até 3.000 mensagens por segundo por ação de API e para reduzir custos. Para a `ReceiveMessage` API, defina o `MaxNumberofMessages` parâmetro para receber até dez mensagens por transação. Para ter mais informações, consulte Ações em lote do Amazon SQS.
- Para filas FIFO com alta taxa de transferência, siga as recomendações para otimizar a utilização da partição. Envie mensagens com os mesmos IDs de grupo de mensagens em lotes. Exclua mensagens ou altere os valores de tempo limite de visibilidade da mensagem em lotes com identificadores de recebimento das mesmas solicitações de `ReceiveMessage` API.
- Aumente o número de `MessageGroupId`valores exclusivos. Isso permite uma distribuição uniforme entre as partições de fila FIFO. Para ter mais informações, consulte Usar o ID do grupo de mensagens do Amazon SQS.

Para obter mais informações, consulte Por que minha fila FIFO do Amazon SQS não retorna todas as mensagens ou mensagens em outros grupos de mensagens? no Guia do Centro de AWS Conhecimento.

Solucionar problemas de mensagens não retornadas para uma chamada de API do Amazon ReceiveMessage SQS

Os tópicos a seguir abordam as causas mais comuns pelas quais uma mensagem do Amazon SQS pode não ser retornada aos consumidores e como solucioná-las. Para obter mais informações, consulte Por que não consigo receber mensagens da minha fila do Amazon SQS? no Guia do Centro de AWS Conhecimento.

Tópicos

- Fila vazia

- Em voo, o limite foi atingido
- Atraso na mensagem
- A mensagem está em andamento
- Método de pesquisa

Fila vazia

Para determinar se uma fila está vazia, use uma sondagem longa para chamar a `ReceiveMessage`API. Você também pode usar as `ApproximateNumberOfMessagesDelayed` CloudWatch métricas `ApproximateNumberOfMessagesVisible`, `ApproximateNumberOfMessagesNotVisible`, e. Se todos os valores métricos forem definidos como 0 por vários minutos, a fila será considerada vazia.

Em voo, o limite foi atingido

Se você usar uma sondagem longa e se o limite de espera da fila (20.000 para FIFO, 120000 para o padrão por padrão) for violado, o Amazon SQS não retornará mensagens de erro que excedam os limites da cota.

Atraso na mensagem

Se a fila do Amazon SQS estiver configurada como uma fila de atraso ou se as mensagens tiverem sido enviadas com temporizadores de mensagens, as mensagens não ficarão visíveis até que o tempo de atraso termine. Para verificar se uma fila está configurada como fila de atraso, use o **`DelaySeconds`** atributo da **`GetQueueAttributes`**API ou do console de fila em Atraso de entrega. Verifique a `ApproximateNumberOfMessagesDelayed` CloudWatch métrica para entender se alguma mensagem está atrasada.

A mensagem está em andamento

Se um consumidor diferente tiver pesquisado a mensagem, ela ficará ativa ou invisível durante o período de tempo limite de visibilidade. As pesquisas adicionais podem retornar um recibo vazio. Verifique a CloudWatch métrica ApproximateNumberOfMessagesVisível para entender o número de mensagens que estão disponíveis para serem recebidas. No caso de filas FIFO, se uma mensagem com o ID do grupo de mensagens estiver em andamento, nenhuma outra mensagem

será retornada, a menos que você exclua a mensagem ou ela fique visível. Isso ocorre porque a ordem das mensagens é mantida no nível do grupo de mensagens em uma fila FIFO.

Método de pesquisa

Se você estiver usando uma sondagem curta, (WaitTimesegundos é 0), o Amazon SQS faz uma amostra de um subconjunto de seus servidores e retorna mensagens somente desses servidores. Portanto, você pode não receber as mensagens, mesmo que elas estejam disponíveis para serem recebidas. As solicitações de pesquisa subsequentes retornarão as mensagens.

Se você estiver usando uma sondagem longa, o Amazon SQS pesquisa todos os servidores e envia uma resposta após coletar pelo menos uma mensagem disponível e até o número máximo especificado. Se o valor de ReceiveMessage WaitTimeSegundos for muito baixo, talvez você não receba todas as mensagens disponíveis.

Solucionar problemas de erros de rede do Amazon SQS

Os tópicos a seguir abordam as causas mais comuns de problemas de rede no Amazon SQS e como solucioná-los.

Tópicos

- ETIMEOUT error
- UnknownHostException error

ETIMEOUT error

O ETIMEOUT erro ocorre quando o cliente não consegue estabelecer uma conexão TCP com um endpoint do Amazon SQS.

Solucionar problemas:

- Verifique a conexão de rede

  Teste sua conexão de rede com o Amazon SQS executando comandos como. `telnet`

  Example: telnet sqs.us-east-1.amazonaws.com 443
- Verifique as configurações de rede

  - Certifique-se de que suas regras de firewall, rotas e listas de controle de acesso (ACLs) locais permitam tráfego na porta que você usa.
  - As regras de saída (saída) do grupo de segurança devem permitir o tráfego para a porta 80 ou 443.
  - As regras de saída (saída) da ACL de rede devem permitir o tráfego para a porta TCP 80 ou 443.
  - As regras de entrada (entrada) da ACL de rede devem permitir o tráfego nas portas TCP 1024-65535.
  - As instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que se conectam à Internet pública devem ter conectividade com a Internet.
- Endpoints da Amazon Virtual Private Cloud (Amazon VPC)

  Se você acessar o Amazon SQS por meio de um endpoint do Amazon VPC, o grupo de segurança dos endpoints deverá permitir tráfego de entrada para o grupo de segurança do cliente na porta 443. A rede ACL associada à sub-rede do VPC endpoint deve ter esta configuração:
  - As regras de saída (saída) da ACL de rede devem permitir o tráfego nas portas TCP 1024-65535 (portas efêmeras).
  - As regras de entrada (entrada) da ACL de rede devem permitir o tráfego na porta 443.

Além disso, a política de endpoint VPC AWS Identity and Access Management (IAM) do Amazon SQS deve permitir o acesso. O exemplo de política de endpoint VPC a seguir especifica que o usuário *MyUser* do IAM tem permissão para enviar mensagens para a fila do Amazon SQS. *MyQueue* Outras ações, usuários do IAM e recursos do Amazon SQS têm acesso negado por meio do VPC endpoint.

```
{
    "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

O UnknownHostException erro ocorre quando o endereço IP do host não pôde ser determinado.

Solucionar problemas:

Use o nslookup utilitário para retornar o endereço IP associado ao nome do host:

- Windows and Linux OS

```
nslookup sqs.<region>.amazonaws.com
```

- AWS CLI ou SDK para endpoints legados do Python:

```
nslookup <region>.queue.amazonaws.com
```

Se você recebeu uma saída malsucedida, siga as instruções em Como o DNS funciona e como soluciono falhas parciais ou intermitentes de DNS? no Guia do Centro de AWS Conhecimento.

Se você recebeu uma saída válida, é provável que seja um problema no nível do aplicativo. Para resolver problemas no nível do aplicativo, tente os seguintes métodos:

- Reinicie seu aplicativo.
- Confirme se seu aplicativo Java não tem um cache DNS inválido. Se possível, configure seu aplicativo para aderir ao TTL do DNS. Para obter mais informações, consulte Configurando o TTL da JVM para pesquisas de nomes DNS.

Para obter informações adicionais sobre como solucionar erros de rede, consulte Como soluciono erros de conexão “ETIMEOUT” e UnknownHost “Exception” do Amazon SQS? no Guia do Centro de AWS Conhecimento.

Solução de problemas de filas do Amazon Simple Queue Service usando o AWS X-Ray

AWS X-Ray coleta dados sobre solicitações que seu aplicativo atende e permite que você visualize e filtre dados para identificar possíveis problemas e oportunidades de otimização. Para qualquer solicitação rastreada para seu aplicativo, você pode ver informações detalhadas sobre a solicitação,

a resposta e as chamadas que seu aplicativo faz para AWS recursos downstream, microsserviços, bancos de dados e APIs HTTP da web.

Para enviar cabeçalhos de AWS X-Ray rastreamento por meio do Amazon SQS, você pode fazer o seguinte:

- Usar o cabeçalho de rastreamento `X-Amzn-Trace-Id`.
- Usar o atributo do sistema de mensagens `AWSTraceHeader`

Para coletar dados sobre erros e latência, é necessário instrumentar o cliente do `AmazonSQS` usando o SDK do AWS X-Ray.

Você pode usar o AWS X-Ray console para visualizar o mapa de conexões entre o Amazon SQS e outros serviços que seu aplicativo usa. Também é possível usar o console para visualizar métricas como a latência média e as taxas de falha. Para obter mais informações, consulte Amazon SQS e AWS X-Rayno Guia do desenvolvedor do AWS X-Ray .

Segurança no Amazon SQS

Esta seção fornece informações sobre a segurança, a autenticação e o controle de acesso do Amazon SQS, bem como a Linguagem de políticas de acesso do Amazon SQS.

Tópicos

Proteção de dados no Amazon SQS

O modelo de responsabilidade AWS compartilhada se aplica à proteção de dados no Amazon Simple Queue Service. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Perguntas Frequentes sobre Privacidade de Dados.. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS LGPD e Modelo de Responsabilidade Compartilhada no AWS Blog de Segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.

- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Amazon SQS ou outros Serviços da AWS usando o console, a API ou AWS os AWS CLI SDKs. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

As seções a seguir fornecem informações sobre a proteção de dados no Amazon SQS.

Tópicos

Criptografia de dados no Amazon SQS

Proteção de dados refere-se a proteger os dados em trânsito (à medida que são transferidos para e do Amazon SQS) e em repouso (enquanto estão armazenados em discos em datacenters do Amazon SQS). Você pode proteger os dados em trânsito usando Secure Sockets Layer (SSL) ou criptografia no lado do cliente. Por padrão, o Amazon SQS armazena mensagens e arquivos usando criptografia de disco. Você pode proteger dados em repouso solicitando que o Amazon SQS criptografe suas mensagens antes de salvá-las no sistema de arquivos criptografados em seus datacenters. O Amazon SQS recomenda o uso do SSE para otimizar a criptografia de dados.

Tópicos

- Gerenciamento de chaves do Amazon SQS

Criptografia em repouso no Amazon SQS

A criptografia no lado do servidor (SSE) permite que você transmita dados sigilosos em filas criptografadas. O SSE protege o conteúdo das mensagens em filas usando chaves de criptografia gerenciadas pelo SQS (SSE-SQS) ou chaves gerenciadas no (SSE-KMS). AWS Key Management Service Para obter informações sobre como gerenciar o SSE usando o AWS Management Console, consulte o seguinte:

- Configurar o SSE-SQS para uma fila (console)
- Configurar o SSE-KMS para uma fila (console)

Para obter informações sobre como gerenciar o SSE usando as AWS SDK for Java (e `GetQueueAttributes` as ações `CreateQueueSetQueueAttributes`, e), consulte os exemplos a seguir:

- Usando criptografia do lado do servidor com filas do Amazon SQS
- Configurando permissões do KMS para Serviços da AWS

A SSE criptografa mensagens assim que o Amazon SQS as recebe. As mensagens são armazenadas no formato criptografado e o Amazon SQS as descriptografa apenas quando elas são enviadas a um consumidor autorizado.

> **Important**
>
> Todas as solicitações para filas com SSE habilitada devem usar HTTPS e o Signature versão 4.
> Uma fila criptografada que usa a chave padrão (chave KMS AWS gerenciada para Amazon SQS) não pode invocar uma função Lambda em outra. Conta da AWS
> Alguns recursos dos AWS serviços que podem enviar notificações para o Amazon SQS usando a AWS Security Token Service `AssumeRole` ação são compatíveis com o SSE, mas funcionam somente com filas padrão:
>
> - Ganchos do ciclo de vida do Auto Scaling
> - AWS Lambda Dead Letter Queues

Para obter informações sobre a compatibilidade de outros produtos com filas criptografadas, consulte Configurar permissões KMS para serviços AWS e a documentação do seu produto.

AWS KMS combina hardware e software seguros e de alta disponibilidade para fornecer um sistema de gerenciamento de chaves dimensionado para a nuvem. Quando você usa o Amazon SQS com AWS KMS, as chaves de dados que criptografam os dados da sua mensagem também são criptografadas e armazenadas com os dados que elas protegem.

Os benefícios de usar o AWS KMS são os seguintes:

- É possível criar e gerenciar AWS KMS keys por conta própria:
- Você também pode usar a chave KMS AWS gerenciada para o Amazon SQS, que é exclusiva para cada conta e região.
- Os padrões AWS KMS de segurança podem ajudá-lo a atender aos requisitos de conformidade relacionados à criptografia.

Para obter mais informações, consulte O que é o AWS Key Management Service? no Guia do desenvolvedor do AWS Key Management Service .

Tópicos

Escopo de criptografia

A SSE criptografa o corpo de uma mensagem em uma fila do Amazon SQS.

A SSE não criptografa o seguinte:

- Metadados de fila (nome e atributos da fila)
- Metadados de mensagens (ID de mensagem, carimbo de data/hora e atributos)
- Métricas por fila

A criptografia de uma mensagem torna indisponível seu conteúdo para usuários não autorizados ou anônimos. Com a SSE habilitada, as solicitações anônimas `SendMessage` e `ReceiveMessage` à fila criptografada serão rejeitadas. As práticas recomendadas de segurança do Amazon SQS não aconselham o uso de solicitações anônimas. Se você quiser enviar solicitações anônimas a uma fila do Amazon SQS, desabilite o SSE. Isso não afeta o funcionamento normal do Amazon SQS:

- Uma mensagem só será criptografada se for enviada após a habilitação da criptografia de uma fila. O Amazon SQS não criptografa mensagens com lista de pendências.
- Qualquer mensagem criptografada permanecerá dessa forma mesmo se a criptografia de sua fila for desabilitada.

A transferência de uma mensagem para uma dead letter queue não afeta sua criptografia:

- Quando o Amazon SQS move uma mensagem de uma fila de origem criptografada para uma fila de mensagens não entregues não criptografada, a mensagem permanece criptografada.
- Quando o Amazon SQS move uma mensagem de uma fila de origem não criptografada para uma fila de mensagens não entregues criptografada, a mensagem permanece descriptografada.

Principais termos

Os seguintes termos-chave podem ajudar você a entender melhor a funcionalidade da SSE. Para obter descrições detalhadas, consulte a Referência da API do Amazon Simple Queue Service.

Chave de dados

A chave (DEK) responsável por criptografar o conteúdo de mensagens do Amazon SQS.

Para obter mais informações, consulte Chaves de dados no Guia do desenvolvedor do AWS Key Management Service no Guia do desenvolvedor do AWS Encryption SDK .

Período de reutilização de chaves de dados

O período de tempo, em segundos, durante o qual o Amazon SQS pode reutilizar uma chave de dados para criptografar ou descriptografar mensagens antes de ligar novamente. AWS KMS Um número inteiro que representa segundos, entre 60 segundos (1 minuto) e 86.400 segundos (24 horas). O padrão é 300 (5 minutos). Para ter mais informações, consulte Entender o período de reutilização de chaves de dados.

> **Note**
>
> No caso improvável de não conseguir acessar AWS KMS, o Amazon SQS continua usando a chave de dados em cache até que a conexão seja restabelecida.

ID da chave do KMS

O alias, o ARN do alias, o ID da chave ou o ARN da chave de uma chave KMS AWS gerenciada ou de uma chave KMS personalizada — na sua conta ou em outra conta. Embora o alias da chave KMS AWS gerenciada para o Amazon SQS seja `alias/aws/sqs` sempre, o alias de uma chave KMS personalizada pode, por exemplo, ser. `alias/MyAlias` Você pode usar essas chaves do KMS para proteger as mensagens em filas do Amazon SQS.

> **Note**
>
> Lembre-se do seguinte:
>
> - Se você não especificar uma chave KMS personalizada, o Amazon SQS usa a chave KMS gerenciada para AWS o Amazon SQS.
> - A primeira vez que você usa o AWS Management Console para especificar a chave KMS AWS gerenciada para o Amazon SQS para uma fila AWS KMS , cria a chave KMS gerenciada para AWS o Amazon SQS.
> - Como alternativa, na primeira vez em que você usa a `SendMessageBatch` ação `SendMessage` or em uma fila com o SSE ativado, AWS KMS cria a chave KMS AWS gerenciada para o Amazon SQS.

Você pode criar chaves KMS, definir as políticas que controlam como as chaves KMS podem ser usadas e auditar o uso da chave KMS usando a seção Chaves gerenciadas pelo cliente do AWS KMS console ou da ação. `CreateKey` AWS KMS Para obter mais informações, consulte Chaves do KMS e Como criar chaves do KMS no Guia do desenvolvedor da AWS Key Management Service . Para obter mais exemplos de identificadores de chave KMS, consulte a KeyIdReferência da AWS Key Management Service API. Para obter informações sobre como encontrar identificadores de chave do KMS, consulte Encontrar o ID da chave e o ARN no Guia do desenvolvedor do AWS Key Management Service .

> **Important**
>
> Há taxas adicionais pelo uso AWS KMS. Para obter mais informações, consulte Estimando custos AWS KMS e Definição de preço do AWS Key Management Service.

Criptografia de envelope

A segurança dos dados criptografados depende em parte da proteção da chave de dados que pode descriptografá-los. O Amazon SQS usa a chave do KMS para criptografar a chave de dados. Em seguida, a chave de dados criptografada é armazenada com a mensagem criptografada. Essa prática de uso de uma chave do KMS para criptografar chaves de dados é conhecida como criptografia de envelope.

Para obter mais informações, consulte Criptografia de envelope no Guia do desenvolvedor do AWS Encryption SDK .

Gerenciamento de chaves do Amazon SQS

O Amazon SQS se integra ao AWS Key Management Service (KMS) para gerenciar chaves KMS para criptografia do lado do servidor (SSE). Consulte Criptografia em repouso no Amazon SQS para obter informações sobre SSE e definições de gerenciamento de chaves. O Amazon SQS usa chaves do KMS para validar e proteger as chaves de dados que criptografam e descriptografam as mensagens. As seções a seguir fornecem informações sobre como trabalhar com chaves do KMS e chaves de dados no produto Amazon SQS.

Tópicos

- Configurar permissões do AWS KMS
- Entender o período de reutilização de chaves de dados
- Estimando custos AWS KMS
- AWS KMS erros

Configurar permissões do AWS KMS

Cada chave do KMS deve ter uma política de chaves. Observe que você não pode modificar a política de chaves de uma chave KMS AWS gerenciada para o Amazon SQS. A política dessa chave

do KMS inclui permissões de uso das filas criptografadas para todos os principais na conta (que estão autorizados a usar o Amazon SQS).

Para uma chave do KMS gerenciada pelo cliente, é necessário configurar a política de chave a fim de adicionar permissões para cada produtor e consumidor de fila. Para fazer isso, nomeie o produtor e o consumidor como usuários na política de chave do KMS. Para obter mais informações sobre AWS KMS permissões, consulte a referência de AWS KMS recursos e operações ou permissões de AWS KMS API no Guia do AWS Key Management Service desenvolvedor.

Como alternativa, é possível especificar as permissões necessárias em uma política do IAM atribuída aos principais que produzem e consomem mensagens criptografadas. Para obter mais informações, consulte Usar políticas do IAM com AWS KMS no Guia do desenvolvedor do AWS Key Management Service .

> **Note**
>
> Embora você possa configurar permissões globais para enviar e receber do Amazon SQS, é AWS KMS necessário nomear explicitamente o ARN completo das chaves KMS em regiões específicas na seção de uma política do IAM. `Resource`

Configurar permissões KMS para serviços AWS

Vários AWS serviços atuam como fontes de eventos que podem enviar eventos para as filas do Amazon SQS. Para permitir que essas fontes de eventos funcionem com filas criptografadas, você deve criar uma chave KMS gerenciada pelo cliente e adicionar permissões na política de chaves para que o serviço use os métodos de AWS KMS API necessários. Execute as etapas a seguir para configurar as permissões.

> **Warning**
>
> Ao alterar a chave KMS para criptografar suas mensagens do Amazon SQS, esteja ciente de que as mensagens existentes criptografadas com a chave KMS antiga permanecerão criptografadas com essa chave. Para descriptografar essas mensagens, você deve reter a chave KMS antiga e garantir que sua política de chaves conceda ao Amazon SQS as permissões para e. `kms:Decrypt` `kms:GenerateDataKey` Depois de atualizar para uma nova chave KMS para criptografar novas mensagens, certifique-se de que todas

as mensagens existentes criptografadas com a chave KMS antiga sejam processadas e removidas da fila antes de excluir ou desativar a chave KMS antiga.

1. Para criar uma chave do KMS gerenciada pelo cliente. Para obter mais informações, consulte Criação de chaves no Guia do desenvolvedor AWS Key Management Service .
2. Para permitir que a fonte do evento de AWS serviço use os métodos `kms:GenerateDataKey` e `kms:Decrypt` da API, adicione a seguinte declaração à política de chaves do KMS.

```
{
   "Version": "2012-10-17",
      "Statement": [{
         "Effect": "Allow",
         "Principal": {
            "Service": "service.amazonaws.com"
         },
         "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
         ],
         "Resource": "*"
       }]
}
```

Substitua “service” no exemplo acima pelo nome do serviço da origem de evento. As origens de evento incluem os serviços a seguir.

Origem do evento.	Nome do serviço
CloudWatch Eventos da Amazon	`events.amazonaws.com`
Notificações de eventos do Amazon S3	`s3.amazonaws.com`
Assinaturas de tópicos do Amazon SNS	`sns.amazonaws.com`

3. Configure uma fila com SSE existente usando o ARN de sua chave do KMS.
4. Forneça o ARN da fila criptografada para a fonte do evento.

Configurar AWS KMS permissões para produtores

Quando o período de reutilização da chave de dados expirar, a próxima chamada do produtor para `SendMessage` ou `SendMessageBatch` também acionará chamadas para `kms:GenerateDataKey` e `kms:Decrypt`. A chamada para `kms:Decrypt` tem o intuito de verificar a integridade da nova chave de dados antes de usá-la. Portanto, o produtor deve ter as permissões `kms:GenerateDataKey` e `kms:Decrypt` para a chave do KMS.

Adicione a declaração a seguir à política do IAM do produtor. Lembre-se de usar os valores de ARN corretos para o recurso da chave e o recurso da fila.

```
{
   "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"
      }]
}
```

Configurar AWS KMS permissões para consumidores

Quando o período de reutilização da chave de dados expirar, a próxima chamada do consumidor `ReceiveMessage` também acionará uma chamada para `kms:Decrypt`, a fim de verificar a integridade da nova chave de dados antes de usá-la. Portanto, o consumidor deve ter a permissão `kms:Decrypt` para qualquer chave do KMS que é usada para criptografar as mensagens na fila específica. Se a fila agir como uma dead letter queue, o consumidor também deverá ter a permissão `kms:Decrypt` para qualquer chave do KMS que for usada para criptografar as mensagens na fila de origem. Adicione a declaração a seguir à política do IAM do consumidor. Lembre-se de usar os valores de ARN corretos para o recurso da chave e o recurso da fila.

```
{
   "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 permissões com proteção delegada confusa

Quando a entidade principal em uma declaração de política é uma AWS entidade principal do serviço da , você pode usar as chaves de condição global `aws:SourceArn` ou `aws:SourceAccount` para proteção contra o cenário de representante confuso. Para usar essas chaves de condição, defina o valor como o nome do recurso da Amazon (ARN) do recurso que está sendo criptografado. Se você não conhece o ARN do recurso, use `aws:SourceAccount` em vez disso.

Nesta política de chaves do KMS, um recurso específico do serviço que é de propriedade da conta `111122223333` tem permissão para chamar o KMS para ações `Decrypt` e `GenerateDataKey`, que ocorrem durante o uso do SSE do 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"
    ]
   }
  }
 }]
}
```

Ao usar filas do Amazon SQS habilitadas para SSE, os seguintes serviços são compatíveis com `aws:SourceArn`:

- Amazon SNS
- Amazon S3
- CloudWatch Eventos
- AWS Lambda
- CodeBuild
- Amazon Connect Customer Profiles
- AWS Auto Scaling
- Amazon Chime

Entender o período de reutilização de chaves de dados

O período de reutilização da chave de dados define a duração máxima de reutilização da mesma chave de dados pelo Amazon SQS. Quando o período de reutilização da chave de dados terminar, o Amazon SQS gerará uma nova chave de dados. Observe as diretrizes a seguir sobre o período de reutilização.

- Um período de reutilização mais curto oferece melhor segurança, mas resulta em mais chamadas para AWS KMS, o que pode gerar cobranças além do nível gratuito.
- Embora a chave de dados seja armazenada em cache separadamente para a criptografia e a descriptografia, o período de reutilização se aplica a ambas as cópias da chave de dados.
- Quando o período de reutilização da chave de dados termina, a próxima chamada `SendMessage` ou `SendMessageBatch` normalmente aciona uma chamada para o AWS KMS `GenerateDataKey` método para obter uma nova chave de dados. Além disso, as próximas

chamadas para `SendMessage` e cada uma `ReceiveMessage` acionará uma chamada AWS KMS `Decrypt` para verificar a integridade da chave de dados antes de usá-la.

- Diretores (Contas da AWS ou usuários) não compartilham chaves de dados (mensagens enviadas por diretores exclusivos sempre recebem chaves de dados exclusivas). Portanto, o volume de chamadas para AWS KMS é um múltiplo do número de principais exclusivos em uso durante o período de reutilização da chave de dados.

Estimando custos AWS KMS

Para prever custos e entender melhor sua AWS fatura, talvez você queira saber com que frequência o Amazon SQS usa sua chave KMS.

> **Note**
>
> Embora a fórmula a seguir possa dar a você uma boa ideia sobre os custos esperados, os custos reais poderão ser mais altos por conta da natureza distribuída do Amazon SQS.

Para calcular o número de solicitações de APIs (`R`) por fila, use a seguinte fórmula:

```
R = (B / D) * (2 * P + C)
```

`B` é o período de faturamento (em segundos).

`D` é o período de reutilização da chave de dados (em segundos).

`P` é o número de entidades de produção que enviam para a fila do Amazon SQS.

`C` é o número de entidades de consumo que recebem da fila do Amazon SQS.

> **Important**
>
> De modo geral, os principais de produção geram o dobro do custo das entidades principais de consumo. Para ter mais informações, consulte Entender o período de reutilização de chaves de dados.
> Se o produtor e o consumidor tiverem usuários diferentes do , o custo aumentará.

Estes são cálculos de exemplo. Para obter informações sobre a definição de preços, consulte Definição de preços do AWS Key Management Service.

Exemplo 1: cálculo do número de chamadas de AWS KMS API para 2 principais e 1 fila

Este exemplo supõe o seguinte:

- O período de faturamento é de 1 a 31 de janeiro (2.678.400 segundos).
- O período de reutilização de chave de dados é definido como 5 minutos (300 segundos).
- Há 1 fila.
- Há 1 entidade principal de produção e 1 entidade principal de consumo.

```
(2,678,400 / 300) * (2 * 1 + 1) = 26,784
```

Exemplo 2: cálculo do número de chamadas de AWS KMS API para vários produtores e consumidores e duas filas

Este exemplo supõe o seguinte:

- O período de faturamento é de 1 a 28 de fevereiro (2.419.200 segundos).
- O período de reutilização de chave de dados é definido como 24 horas (86.400 segundos).
- Há duas filas.
- A primeira fila tem 3 entidades principais de produção e 1 entidade principal de consumo.
- A segunda fila tem 5 entidades principais de produção e 2 entidades principais de consumo.

```
(2,419,200 / 86,400 * (2 * 3 + 1)) + (2,419,200 / 86,400 * (2 * 5 + 2)) = 532
```

AWS KMS erros

Quando você trabalha com o Amazon SQS e AWS KMS, você pode encontrar erros. As referências a seguir descrevem os erros e possíveis soluções de problemas.

- Erros comuns do AWS KMS
- Erros de descriptografia do AWS KMS
- AWS KMS GenerateDataKey erros

Privacidade do tráfego entre redes no Amazon SQS

Um endpoint da Amazon Virtual Private Cloud (Amazon VPC) para Amazon SQS é uma entidade lógica dentro de uma VPC que permite conectividade apenas com o Amazon SQS. A VPC roteia as solicitações para o Amazon SQS e as respostas de volta para a VPC. As seções a seguir contêm informações sobre como trabalhar com VPC endpoints e criar políticas de VPC endpoint.

Tópicos

Endpoints da Amazon Virtual Private Cloud para o Amazon SQS

Se você usa o Amazon VPC para hospedar seus AWS recursos, você pode estabelecer uma conexão entre seu VPC e o Amazon SQS. Você pode usar essa conexão para enviar mensagens às suas filas do Amazon SQS sem precisar passar pela Internet pública.

A Amazon VPC permite que você lance AWS recursos em uma rede virtual personalizada. Você pode usar uma VPC para controlar as configurações de rede, como o intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Para obter informações sobre como criar suas próprias VPCs, consulte o Guia do usuário da Amazon VPC.

Para conectar a VPC ao Amazon SQS, primeiro você deve definir um endpoint da VPC de interface, que permite conectar a VPC a outros produtos da AWS . O endpoint fornece uma conectividade confiável e escalável ao Amazon SQS sem a necessidade de um gateway da Internet, de uma instância de conversão de endereço de rede (NAT) ou de uma conexão VPN. Para obter mais informações, consulte Tutorial: Envio de uma mensagem a uma fila do Amazon SQS pela Amazon Virtual Private Cloud e Exemplo 5: negar o acesso se não vier de um VPC endpoint neste guia e Endpoints da VPC de interface (AWS PrivateLink) no Guia do usuário da Amazon VPC.

> **Important**
>
> - Você pode usar a Amazon Virtual Private Cloud somente com endpoints HTTPS do Amazon SQS.
> - Ao configurar o Amazon SQS para enviar mensagens pela Amazon VPC, habilite o DNS privado e especifique endpoints no formato `sqs.us-east-2.amazonaws.com`.

- O DNS privado não oferece suporte a endpoints legados, como `queue.amazonaws.com` ou *`us-east-2`*`.queue.amazonaws.com`.

Criar uma política de endpoint da Amazon VPC para o Amazon SQS

É possível criar uma política para endpoints da Amazon VPC para o Amazon SQS na qual se especifica o seguinte:

- A entidade principal que pode executar ações.
- As ações que podem ser executadas.
- Os recursos sobre os quais as ações podem ser realizadas.

Para obter mais informações, consulte Controlar o acesso a produtos com endpoints da VPC no Guia do usuário da Amazon VPC

O exemplo de política de endpoint da VPC a seguir especifica que o usuário `MyUser` tem permissão para enviar mensagens à fila `MyQueue` do 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"
      }
   }]
}
```

O seguinte é negado:

- Outras ações de API do Amazon SQS, como `sqs:CreateQueue` e `sqs:DeleteQueue`.
- Outros usuários e regras do que tentam usar esse VPC endpoint.
- Envio de mensagens de `MyUser` para outra fila do Amazon SQS.

> **Note**
>
> O usuário ainda pode usar outras ações de API do Amazon SQS de fora da VPC. Para ter mais informações, consulte Exemplo 5: negar o acesso se não vier de um VPC endpoint.

Gerenciamento de identidade e acesso no Amazon SQS

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para utilizar os recursos do Amazon SQS. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no Amazon SQS.

Usuário do serviço: se você usar o serviço do Amazon SQS para fazer seu trabalho, o administrador fornecerá as credenciais e as permissões de que você precisa. À medida que mais recursos do Amazon SQS forem usados para realizar o trabalho, talvez sejam necessárias permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não puder acessar um recurso no Amazon SQS, consulte Resolução de problemas de identidade e acesso do Amazon Simple Queue Service.

Administrador do serviço: se você for o responsável pelos recursos do Amazon SQS em sua empresa, provavelmente terá acesso total ao Amazon SQS. Cabe a você determinar quais funcionalidades e recursos do Amazon SQS os usuários do seu serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como sua empresa pode usar o IAM com o Amazon SQS, consulte Como o Amazon Simple Queue Service funciona com o IAM.

Administrador do IAM: se você é um administrador do IAM, talvez queira saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao Amazon SQS. Para visualizar exemplos de políticas baseadas em identidade do Amazon SQS que podem ser usadas no IAM, consulte Melhores práticas de política.

Autenticando com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como usuário do Usuário raiz da conta da AWS IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia AWS IAM Identity Center do usuário e Utilizar a autenticação multifator (MFA) na AWS no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do Usuário do IAM.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte “O que é o Centro de Identidade do IAM?” no Guia do usuário AWS IAM Identity Center .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de um perfil) no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma função do IAM no AWS Management Console trocando de funções. Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para o uso de perfis, consulte Utilizar perfis do IAM no Guia do usuário do IAM.

Funções do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidades de terceiros no Guia do Usuário do IAM. Se você usar o Centro de identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões no Guia do usuário AWS IAM Identity Center .
- Permissões temporárias para usuários do IAM –- um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas –- é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service

(Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

- Função de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

- Aplicativos em execução no Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo AWS CLI solicitações de API. AWS É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Utilizar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar perfis do IAM, consulte Quando criar um perfil do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do Usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissões para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem presumir os perfis.

As políticas do IAM definem permissões para uma ação independente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criando políticas do IAM no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do Usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um atributo avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do Usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em. AWS Organizations AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte How SCPs work (Como os SCPs funcionam) no Guia do usuário do AWS Organizations .
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do

usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do Usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Visão geral do gerenciamento de acesso no Amazon SQS

Cada AWS recurso é de propriedade de um Conta da AWS, e as permissões para criar ou acessar um recurso são regidas por políticas de permissões. Um administrador da conta pode anexar políticas de permissões a identidades do IAM (usuários, grupos e funções) e alguns produtos (como o Amazon SQS) também oferecem suporte à anexação de políticas de permissões aos recursos.

> **Note**
>
> O administrador da conta (ou usuário administrador) é um usuário com privilégios administrativos. Para obter mais informações, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Ao conceder permissões, você especifica os usuários que recebem permissões, o recurso para o qual as permissões são concedidas e as ações específicas que você deseja permitir no recurso.

Tópicos

- Recursos e operações do Amazon Simple Queue Service
- Informações sobre propriedade de recursos
- Gerenciamento de acesso aos recursos
- Especificar elementos da política: ações, efeitos, recursos e entidades principais

Recursos e operações do Amazon Simple Queue Service

No Amazon SQS, o único recurso é a fila. Em uma política, use um nome de recurso da Amazon (ARN) para identificar o recurso ao qual a política se aplica. O seguinte recurso tem um ARN exclusivo associado a ele:

Tipo de recurso	Formato ARN
Fila	`arn:aws:sqs:` *`region`*`:`*`account_id`*`:`*`queue_name`*

Veja a seguir exemplos do formato do ARN para filas:

- Um ARN para uma fila nomeada `my_queue` na região Leste dos EUA (Ohio), pertencente à AWS Conta 123456789012:

```
arn:aws:sqs:us-east-2:123456789012:my_queue
```

- Um ARN para uma fila chamada `my_queue` em cada uma das diferentes regiões compatíveis com o Amazon SQS:

```
arn:aws:sqs:*:123456789012:my_queue
```

- Um ARN que usa `*` ou `?` como um curinga para o nome da fila. No exemplo a seguir, o ARN corresponde a todas as filas prefixadas com `my_prefix_`:

```
arn:aws:sqs:*:123456789012:my_prefix_*
```

Você pode obter o valor do ARN para uma fila existente chamando a ação `GetQueueAttributes`. O valor do atributo `QueueArn` é o ARN da fila. Para obter mais informações sobre ARNs, consulte ARNs do IAM no Guia do usuário do IAM.

O Amazon SQS fornece um conjunto de ações que funcionam com o recurso de fila. Para ter mais informações, consulte Permissões da API do Amazon SQS: referência de ações e recurso.

Informações sobre propriedade de recursos

Ele Conta da AWS possui os recursos criados na conta, independentemente de quem criou os recursos. Mais especificamente, o proprietário do recurso é a Conta da AWS da entidade principal (ou seja, a conta raiz, um usuário ou um perfil do IAM) que autentica a solicitação de criação de recursos. Os seguintes exemplos mostram como isso funciona:

- Se você usar as credenciais da sua conta raiz Conta da AWS para criar uma fila do Amazon SQS, Conta da AWS você é o proprietário do recurso (no Amazon SQS, o recurso é a fila do Amazon SQS).
- Se você criar um usuário no seu Conta da AWS e conceder permissões para criar uma fila para o usuário, o usuário poderá criar a fila. No entanto, a Conta da AWS (à qual o usuário pertence) é a proprietária do recurso de fila.
- Se você criar uma função do IAM Conta da AWS com permissões para criar uma fila do Amazon SQS, qualquer pessoa que possa assumir a função poderá criar uma fila. Seu Conta da AWS (ao qual a função pertence) é proprietário do recurso de fila.

Gerenciamento de acesso aos recursos

Uma política de permissões descreve as permissões concedidas às contas. A seção a seguir explica as opções disponíveis para a criação das políticas de permissões.

> **Note**
>
> Esta seção discute o uso do IAM no contexto do Amazon SQS. Não são fornecidas informações detalhadas sobre o serviço IAM. Para ver a documentação completa do IAM, consulte What is IAM? no IAM User Guide. Para obter mais informações sobre a sintaxe e as descrições da política do IAM, consulte a Referência de política do AWS IAM no Guia do usuário do IAM.

As políticas anexadas a uma identidade do IAM são conhecidas como políticas baseadas em identidade (políticas do IAM;) e as políticas anexadas a um recurso são conhecidas como políticas baseadas em recurso.

Políticas baseadas em identidade

Há duas maneiras de conceder aos usuários permissões às suas filas do Amazon SQS: usando os sistemas de políticas do Amazon SQS e do IAM. Você pode usar um dos sistemas, ou ambos, para anexar políticas a usuários ou funções. Na maioria dos casos, você pode atingir o mesmo resultado usando um dos sistemas. Por exemplo, você pode fazer o seguinte:

- Anexar uma política de permissões a um usuário ou grupo na conta: para conceder a um usuário permissões para criar uma fila do Amazon SQS, anexe uma política de permissões a um usuário ou grupo a que o usuário pertença.
- Anexar uma política de permissões a um usuário em outra Conta da AWS: para conceder a um usuário permissões para criar uma fila do Amazon SQS, anexe uma política de permissões do Amazon SQS a um usuário em outra Conta da AWS.

  As permissões entre contas não se aplicam às seguintes ações:

  - `AddPermission`
  - `CancelMessageMoveTask`
  - `CreateQueue`
  - `DeleteQueue`
  - `ListMessageMoveTask`
  - `ListQueues`
  - `ListQueueTags`
  - `RemovePermission`
  - `SetQueueAttributes`
  - `StartMessageMoveTask`
  - `TagQueue`
  - `UntagQueue`
- Anexar uma política de permissões a uma função (conceder permissões entre contas): para conceder permissões entre contas, anexe uma política de permissões baseada em identidade a uma função do IAM. Por exemplo, o Conta da AWS administrador A pode criar uma função para conceder permissões entre contas a Conta da AWS B (ou a um AWS serviço) da seguinte forma:
  - Um administrador da conta A cria um perfil do IAM e anexa uma política de permissões, que concede permissões em recursos da conta A ao perfil.

- O administrador da conta A anexa uma política de confiança à função que identifica a conta B como a entidade principal, que pode assumir a função.
- O administrador da conta B delega a permissão para assumir a função a qualquer usuário na conta B. Isso permite que os usuários na conta B criem ou acessem filas na conta A.

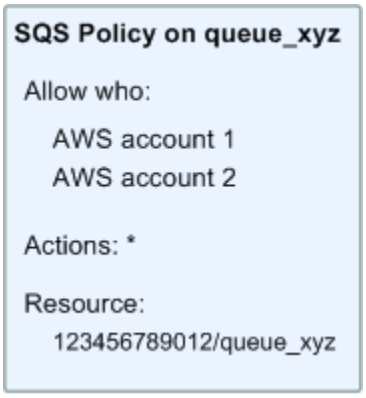
> **Note**
>
> Se você quiser conceder a permissão para assumir a função em um AWS serviço, o principal na política de confiança também pode ser um diretor AWS de serviço.

Para obter mais informações sobre o uso do IAM para delegar permissões, consulte Gerenciamento de acesso no Guia do usuário do IAM.

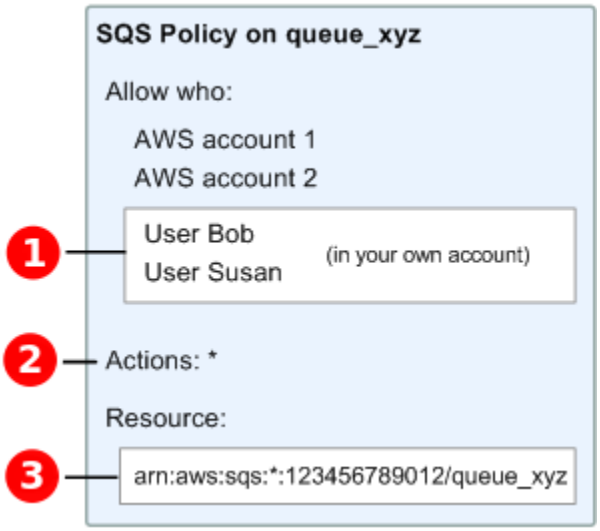
Embora o Amazon SQS funcione com políticas do IAM, ele tem sua própria infraestrutura de políticas. Você pode usar uma política do Amazon SQS com uma fila para especificar quais AWS contas têm acesso à fila. Você pode especificar o tipo de acesso e condições (por exemplo, uma condição que conceda permissões para usar `SendMessage`, `ReceiveMessage` se a solicitação for feita antes de 31 de dezembro de 2010). As ações específicas para as quais você pode conceder permissões são um subconjunto de toda a lista de ações do Amazon SQS. Quando você grava uma política do Amazon SQS e especifica `*` para “permitir todas as ações do Amazon SQS”, significa que um usuário pode realizar todas as ações nesse subconjunto.

O diagrama a seguir ilustra o conceito de uma dessas políticas básicas do Amazon SQS que abrange o subconjunto de ações. A política é para `queue_xyz` e concede à AWS Conta 1 e à AWS Conta 2 permissões para usar qualquer uma das ações permitidas com a fila especificada.

> **Note**
>
> O recurso na política é especificado como`123456789012/queue_xyz`, onde `123456789012` está o AWS ID da conta que possui a fila.



Com a introdução do IAM e os conceitos de usuários e nomes de recursos da Amazon (ARNs), algumas coisas foram alteradas nas políticas do SQS. O diagrama e a tabela a seguir descrevem as alterações.



1 Para obter informações sobre como conceder permissões a usuários em contas diferentes, consulte Tutorial: Delegar acesso entre AWS contas usando funções do IAM no Guia do usuário do IAM.

2 O subconjunto de ações incluídas em * foi expandido. Para obter uma lista de ações permitidas, consulte Permissões da API do Amazon SQS: referência de ações e recurso.

3 Você pode especificar o recurso usando o nome do recurso da Amazon (ARN), a forma padrão de especificar recursos nas políticas do IAM. Para obter informações sobre o formato ARN para filas do Amazon SQS, consulte Recursos e operações do Amazon Simple Queue Service.

Por exemplo, de acordo com a política do Amazon SQS no diagrama anterior, qualquer pessoa que possua as credenciais de segurança da AWS Conta 1 ou AWS da Conta 2 pode acessar. `queue_xyz` Além disso, os usuários Bob e Susan em sua própria conta da AWS (com o ID `123456789012`) podem acessar a fila.

Antes da introdução do IAM, o Amazon SQS concedia automaticamente ao criador de uma fila o controle total sobre ela (ou seja, o acesso a todas as ações possíveis do Amazon SQS nessa fila). Isso não é mais verdadeiro, a menos que o criador use credenciais de segurança da AWS . Qualquer usuário que tenha permissões para criar uma fila também deve ter permissões para usar outras ações do Amazon SQS, para fazer qualquer coisa com as filas criadas.

Veja a seguir uma política de exemplo que permite que um usuário use todas as ações do Amazon SQS, mas apenas com as filas cujos nomes estejam prefixados com a string literal `bob_queue_`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:*:123456789012:bob_queue_*"
   }]
}
```

Para obter mais informações, consulte Usando políticas com o Amazon SQS e Identidades (usuários, grupos e funções) no Guia do usuário do IAM.

Especificar elementos da política: ações, efeitos, recursos e entidades principais

Para cada recurso do Amazon Simple Queue Service, o produto define um conjunto de ações. Para conceder permissões a essas ações, o Amazon SQS define um conjunto de ações que podem ser especificadas em uma política.

> **Note**
>
> A execução de uma ação de pode exigir permissões para mais de uma ação. Ao conceder permissões para ações específicas, você também identifica o recurso para o qual as ações são permitidas ou recusadas.

Estes são os elementos de política mais básicos:

- Recurso: em uma política, você usa um Amazon Resource Name (ARN – Nome de recurso da Amazon) para identificar o recurso a que a política se aplica.
- Ação: você usa palavras-chave de ação para identificar as ações de recurso que deseja permitir ou negar. Por exemplo, a permissão `sqs:CreateQueue` permite que o usuário execute a ação `CreateQueue` do Amazon Simple Queue Service.
- Efeito: você especifica o efeito quando o usuário solicita a ação específica, que pode ser permitir ou negar. Se você não conceder explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar explicitamente o acesso a um recurso a fim de ter certeza de que um usuário não conseguirá acessá-lo, mesmo que uma política diferente conceda acesso.
- Principal: em políticas baseadas em identidade (políticas do IAM), o usuário ao qual a política é anexada é implicitamente o principal. Para as políticas baseadas em recursos, você especifica quais usuários, contas, serviços ou outras entidades deseja que recebam permissões (isso se aplica somente a políticas baseadas em recursos).

Para saber mais sobre a sintaxe e as descrições da política do Amazon SQS, consulte a AWS Referência de política do IAM da no Guia do usuário do IAM.

Para ver uma tabela com todas as ações do Amazon Simple Queue Service e os recursos a que elas se aplicam, consulte Permissões da API do Amazon SQS: referência de ações e recurso.

Como o Amazon Simple Queue Service funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon SQS, entenda que recursos do IAM estão disponíveis para uso com o Amazon SQS.

Recursos do IAM que você pode usar com o Amazon Simple Queue Service

Atributo do IAM	Compatibilidade com o Amazon SQS
Políticas baseadas em identidade	Sim
Políticas baseadas em atributos	Sim
Ações das políticas	Sim
Atributos de políticas	Sim

Atributo do IAM	Compatibilidade com o Amazon SQS
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Sessões de acesso direto (FAS)	Sim
Perfis de serviço	Sim
Perfis vinculados ao serviço	Não

Para obter uma visão de alto nível de como o Amazon SQS e AWS outros serviços funcionam com a maioria dos recursos do IAM, AWS consulte os serviços que funcionam com o IAM no Guia do usuário do IAM.

Controle de acesso

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

> **Note**
>
> É importante entender que todos Contas da AWS podem delegar suas permissões aos usuários em suas contas. O acesso entre contas permite que você compartilhe o acesso aos seus AWS recursos sem precisar gerenciar usuários adicionais. Para obter informações sobre como usar o acesso entre contas, consulte Habilitar o acesso entre contas no Guia do usuário do IAM.

Consulte Limitações das políticas personalizadas do Amazon SQS para obter mais detalhes sobre permissões de conteúdo cruzado e chaves de condição nas políticas personalizadas do Amazon SQS.

Políticas baseadas em identidade do Amazon SQS

Suporta políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criando políticas do IAM no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da política JSON do IAM no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amazon SQS

Para ver exemplos de políticas baseadas em identidade do Amazon SQS, consulte Melhores práticas de política.

Políticas baseadas em recursos no Amazon SQS

É compatível com políticas baseadas em atributos	Sim

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os

administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações de políticas para o Amazon SQS

Oferece compatibilidade com ações de políticas	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista das ações do Amazon SQS, consulte Tipos de recursos definidos pelo Amazon SQS na Referência de autorização do serviço.

As ações de política no Amazon SQS usam o seguinte prefixo antes da ação:

```
sqs
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
      "sqs:action1",
      "sqs:action2"
         ]
```

Para ver exemplos de políticas baseadas em identidade do Amazon SQS, consulte Melhores práticas de política.

Recursos de políticas para o Amazon SQS

Oferece compatibilidade com recursos de políticas	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista de tipos de recurso do Amazon SQS e seus ARNs, consulte Ações definidas pelo Amazon SQS na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Tipos de recursos definidos pelo Amazon SQS.

Para ver exemplos de políticas baseadas em identidade do Amazon SQS, consulte Melhores práticas de política.

Chaves de condição de política para o Amazon SQS

Suporta chaves de condição de política específicas de serviço	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do Amazon SQS, consulte Ações, recursos e chaves de condição do AWS Key Management Service na Referência de autorização do serviço. Para saber com quais ações e recursos você pode usar a chave de condição, consulte Tipos de recursos definidos pelo Amazon SQS.

Para ver exemplos de políticas baseadas em identidade do Amazon SQS, consulte Melhores práticas de política.

ACLs no Amazon SQS

Oferece compatibilidade com ACLs	Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o Amazon SQS

Oferece compatibilidade com ABAC (tags em políticas)	Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações onde o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do Usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Utilizar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Amazon SQS

Oferece compatibilidade com credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS "Trabalhe com o IAM" no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte Alternar para um perfil (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Sessões de acesso direto para o Amazon SQS

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Perfis de serviço para o Amazon SQS

Oferece compatibilidade com funções de serviço	Sim

O perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amazon SQS. Edite perfis de serviço somente quando o Amazon SQS fornecer orientação para isso.

Perfis vinculados ao serviço para o Amazon SQS

Oferece suporte a perfis vinculados ao serviço	Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Atualizações do Amazon SQS para AWS políticas gerenciadas

Para adicionar permissões a usuários, grupos e perfis, é mais fácil usar políticas gerenciadas pela AWS do que gravar políticas por conta própria. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, é possível usar nossas políticas gerenciadas pela AWS . Essas políticas abrangem casos de uso comuns e estão disponíveis na sua conta da AWS . Para obter mais informações sobre políticas AWS gerenciadas, consulte políticas AWS gerenciadas no Guia do usuário do IAM.

AWS os serviços mantêm e atualizam as políticas AWS gerenciadas. Você não pode alterar as permissões nas políticas AWS gerenciadas. Ocasionalmente, os serviços adicionam permissões adicionais a uma política AWS gerenciada para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política AWS gerenciada quando um novo recurso é lançado ou quando novas operações são disponibilizadas. Os serviços não removem as permissões de uma política AWS gerenciada, portanto, as atualizações de políticas não violarão suas permissões existentes.

Além disso, AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política de ReadOnlyacesso AWS gerenciado fornece acesso somente de leitura a todos os AWS serviços e recursos. Quando um serviço lança um novo recurso, AWS adiciona permissões somente de leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de funções de trabalho, consulte Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.

AWS política gerenciada: AmazonSQS FullAccess

É possível anexar a política `AmazonSQSFullAccess` às suas identidades do Amazon SQS. Essa política concede permissões de acesso total ao Amazon SQS.

Para ver as permissões dessa política, consulte AmazonSQS FullAccess na Referência de políticas AWS gerenciadas.

AWS política gerenciada: ReadOnly AmazonSQS Access

É possível anexar a política `AmazonSQSReadOnlyAccess` às suas identidades do Amazon SQS. Essa política concede permissões de acesso somente leitura ao Amazon SQS.

Para ver as permissões dessa política, consulte AmazonSQS ReadOnly Access na Referência de políticas AWS gerenciadas.

Atualizações do Amazon SQS para AWS políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Amazon SQS desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre mudanças nesta página, assine o feed RSS na página Histórico de documentos do Amazon SQS.

Alteração	Descrição	Data
Acesso ao Amazon ReadOnly SQS	O Amazon SQS adicionou uma nova ação que permite listar as tarefas mais recentes de movimentação de mensagens (até dez) em uma fila de origem específica. Essa ação está associada à operação de API `ListMessageMoveTasks`.	9 de junho de 2023

Resolução de problemas de identidade e acesso do Amazon Simple Queue Service

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com o Amazon SQS e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no Amazon SQS
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos do Amazon SQS

Não tenho autorização para executar uma ação no Amazon SQS

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, é preciso atualizar suas políticas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um recurso do *`my-example-widget`* fictício, mas não tem as permissões fictícias do `sqs:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 sqs:GetWidget on resource: my-example-widget
```

Nesse caso, a política de Mateo deve ser atualizada para permitir que ele tenha acesso ao recurso *`my-example-widget`* usando a ação `sqs:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Caso receba uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, as políticas deverão ser atualizadas para permitir a transmissão de um perfil ao Amazon SQS.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Amazon SQS. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos do Amazon SQS

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon SQS é compatível com esses recursos, consulte Como o Amazon Simple Queue Service funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Usando políticas com o Amazon SQS

Este tópico fornece exemplos de políticas baseadas em identidade em que um administrador de conta pode anexar políticas de permissões a identidades do IAM (usuários, grupos e funções).

> **Important**
>
> Recomendamos analisar primeiro os tópicos introdutórios que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos recursos do Amazon Simple Queue

Service. Para ter mais informações, consulte Visão geral do gerenciamento de acesso no Amazon SQS.
Com exceção de `ListQueues`, todas as ações do Amazon SQS oferecem suporte a permissões no nível do recurso. Para ter mais informações, consulte Permissões da API do Amazon SQS: referência de ações e recurso.

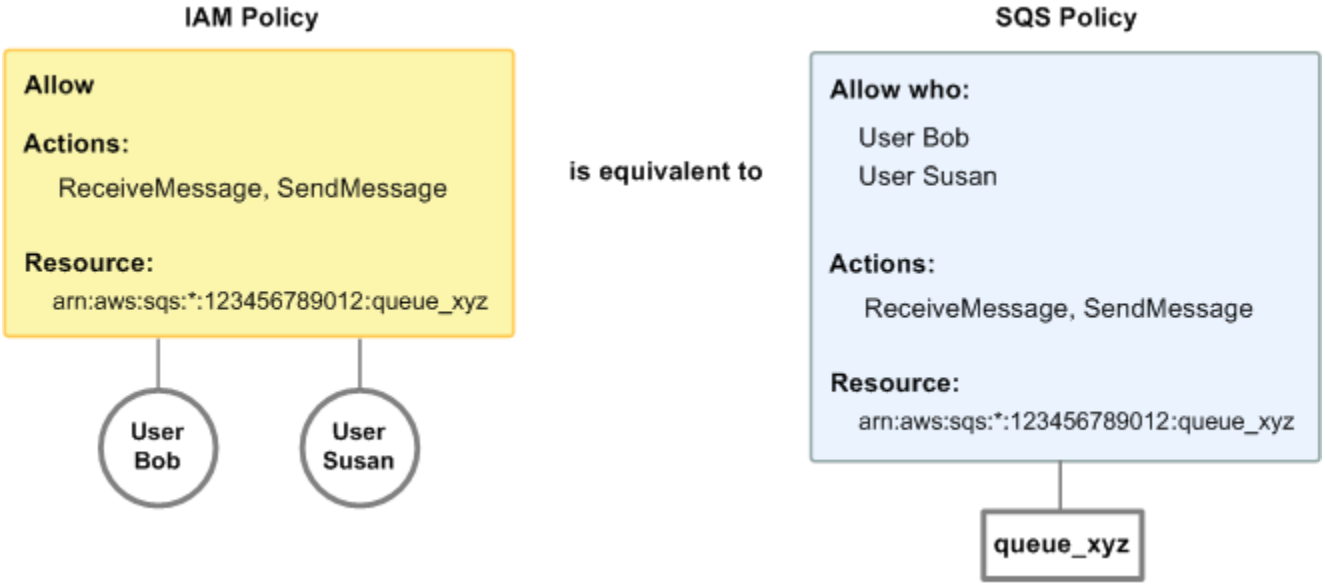
Tópicos

Usar políticas do Amazon SQS e do IAM

Há duas maneiras de conceder aos usuários permissões às filas do Amazon SQS: usando os sistemas de políticas do Amazon SQS e do IAM. Você pode usar um, o outro ou ambos. Na maioria dos casos, você obtém o mesmo resultado com qualquer um deles.

Por exemplo, o diagrama a seguir mostra uma política do IAM e uma política equivalente do Amazon SQS. A política do IAM concede os direitos ao Amazon SQS `ReceiveMessage` e às `SendMessage` ações da fila chamada `queue_xyz` em sua AWS conta, e a política é anexada aos usuários chamados Bob e Susan (Bob e Susan têm as permissões declaradas na política). Essa política do Amazon SQS também oferece a Bob e Susan direitos às ações `ReceiveMessage` e `SendMessage` para a mesma fila.

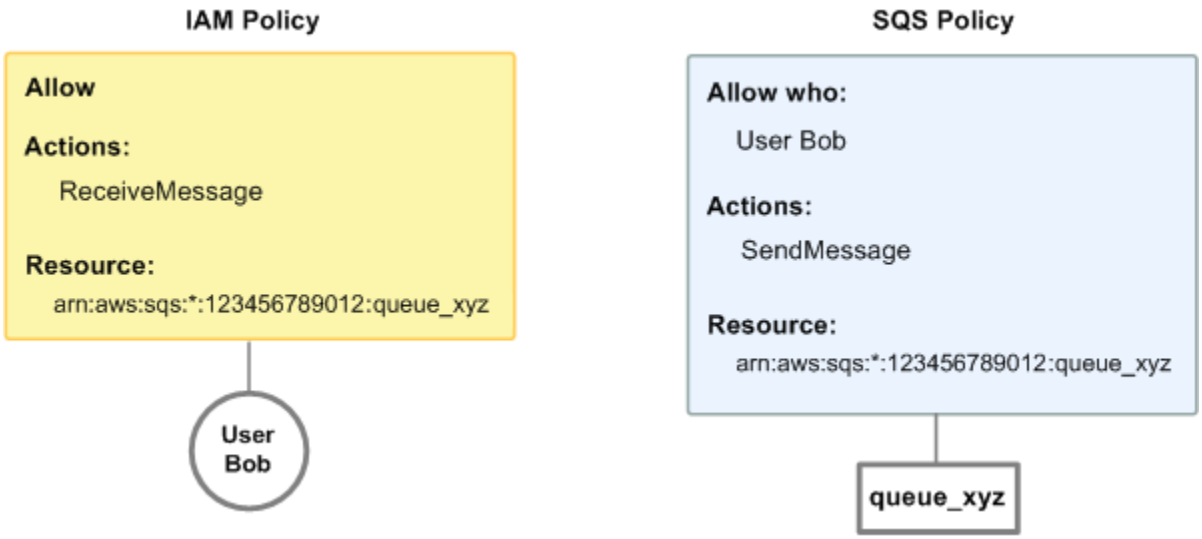
> **Note**
>
> O exemplo a seguir mostra políticas simples sem condições. Você pode especificar uma determinada condição na política e obter o mesmo resultado.



Há uma grande diferença entre as políticas do IAM e do Amazon SQS: o sistema de políticas do Amazon SQS permite que você conceda permissão para AWS outras contas, enquanto o IAM não.

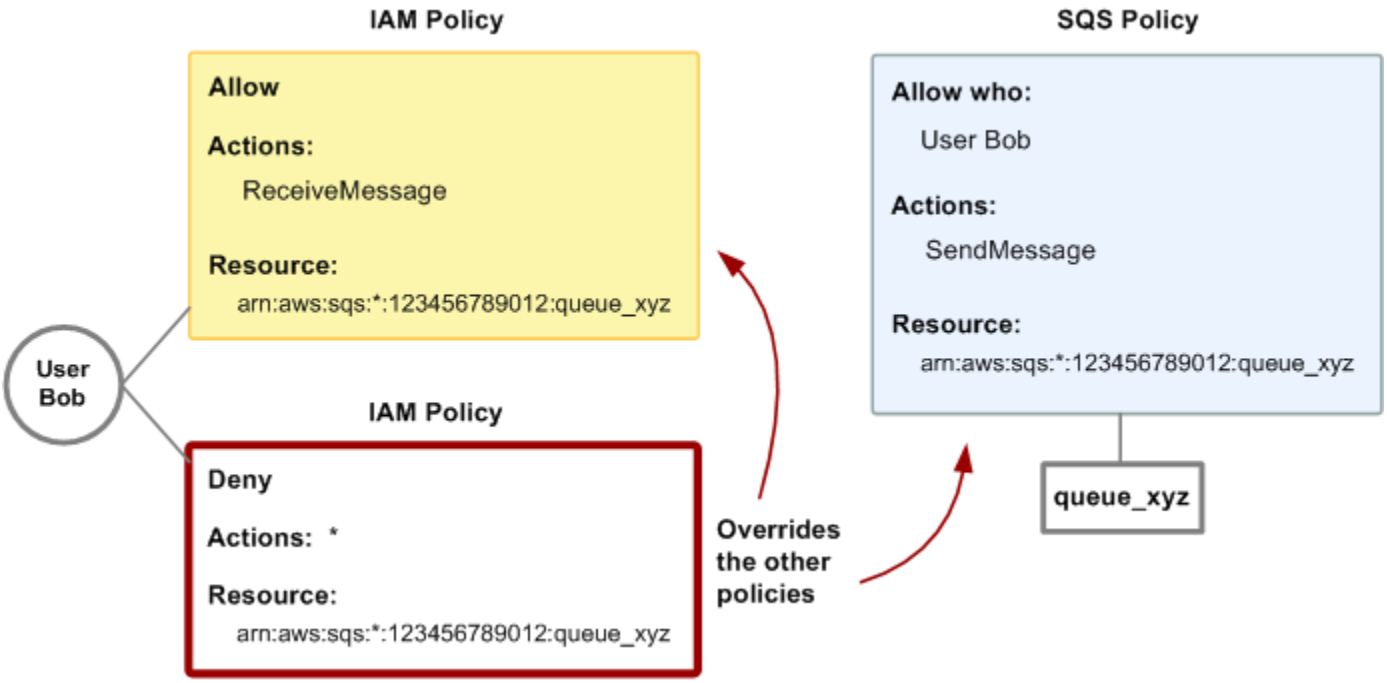
Você é quem decide como usar os dois sistemas para gerenciar suas permissões. Os exemplos a seguir mostram como os dois sistemas de política funcionam em conjunto.

- No primeiro exemplo, Bob tem uma política do IAM e uma do Amazon SQS que se aplicam à sua conta. A política do IAM concede à sua conta permissão para a ação `ReceiveMessage` em `queue_xyz`, enquanto a política do Amazon SQS fornece à sua conta permissão para a ação `SendMessage` na mesma fila. O seguinte diagrama ilustra o conceito.



Se Bob enviar uma solicitação `ReceiveMessage` a `queue_xyz`, a política do IAM permitirá a ação. Se Bob enviar uma solicitação `SendMessage` a `queue_xyz`, a política do Amazon SQS permitirá a ação.

- No segundo exemplo, Bob abusa de seu acesso a `queue_xyz`, para que seja necessário remover todo o seu acesso à fila. O mais fácil a fazer é adicionar uma política que negue a ele acesso a todas as ações para a fila. Essa política substitui as outras duas, pois uma `deny` explícita sempre substitui uma `allow`. Para obter mais informações sobre a lógica de avaliação da política, consulte Usar políticas personalizadas com linguagem de políticas de acesso do Amazon SQS. O seguinte diagrama ilustra o conceito.



Você também pode adicionar outra instrução à política do Amazon SQS que nega a Bob qualquer tipo de acesso à fila. Ela tem o mesmo efeito que a adição de uma política do IAM que nega o acesso de Bob à fila. Para obter exemplos de políticas que abranjam ações e recursos do Amazon SQS, consulte Exemplos básicos de políticas do Amazon SQS. Para obter mais informações sobre como elaborar políticas do Amazon SQS, consulte Usar políticas personalizadas com linguagem de políticas de acesso do Amazon SQS.

Permissões necessárias para usar o console do Amazon SQS

Um usuário que queira trabalhar com o console do Amazon SQS deve ter o conjunto mínimo de permissões para trabalhar com as filas do Amazon SQS na Conta da AWS do usuário. Por exemplo,

o usuário deve ter a permissão para chamar a ação `ListQueues` a fim de poder listar as filas, ou a permissão para chamar a ação `CreateQueue` para poder criar filas. Além de permissões do Amazon SQS, para inscrever uma fila do Amazon SQS em um tópico do Amazon SNS, o console também exige permissões para ações do Amazon SNS.

Se você criar uma política do IAM que seja mais restritiva que as permissões mínimas necessárias, o console poderá não funcionar como pretendido para os usuários com essa política do IAM.

Você não precisa permitir permissões mínimas do console para usuários que fazem chamadas somente para as ações AWS CLI ou para as ações do Amazon SQS.

Exemplos de políticas baseadas em identidade para o Amazon SQS

Por padrão, os usuários e perfis não têm permissão para criar ou modificar recursos do Amazon SQS. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder aos usuários permissão para executar ações nos recursos de que precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem presumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte Criação de políticas do IAM no Guia do Usuário do IAM.

Para obter detalhes sobre ações e tipos de recurso definidos pelo Amazon SQS, incluindo o formato dos ARNs para cada um dos tipos de recurso, consulte Ações, recursos e chaves de condição do AWS Key Management Service na Referência de autorização do serviço.

> **Note**
>
> Ao configurar ganchos do ciclo de vida para o Amazon EC2 Auto Scaling, não é necessário escrever uma política para enviar mensagens a uma fila do Amazon SQS. Para obter mais informações, consulte Amazon EC2 Auto Scaling Lifecycle Hooks no Guia do usuário do Amazon EC2.

Tópicos

- Melhores práticas de política
- Usar o console do Amazon SQS
- Permitir que usuários visualizem suas próprias permissões
- Permitir que um usuário crie filas

- Permitir que os desenvolvedores escrevam mensagens em uma fila compartilhada
- Permita que os gerentes obtenham o tamanho geral das filas
- Permitir que um parceiro envie mensagens para uma fila específica

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon SQS em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do Usuário do IAM.
- Aplique permissões de privilégio mínimo –- ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do Usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso –- você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode gravar uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: Condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais –- o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de 100 verificações de política e recomendações acionáveis para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.

- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas Recomendadas de Segurança no IAM no Guia do Usuário do IAM.

Usar o console do Amazon SQS

Para acessar o console do Amazon Simple Queue Service, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amazon SQS em seu. Conta da AWS Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam a operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o console do Amazon SQS, anexe também a política gerenciada do Amazon `AmazonSQSReadOnlyAccess` AWS SQS às entidades. Para obter mais informações, consulte Adicionando Permissões a um Usuário no Guia do Usuário do IAM.

Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
        }
    ]
}
```

Permitir que um usuário crie filas

No exemplo a seguir, criamos uma política para Bob que permite acessar todas as ações do Amazon SQS, mas apenas com as filas cujos nomes sejam prefixados com a string literal `alice_queue_`.

O Amazon SQS não concede automaticamente ao criador de uma fila permissões para usá-la. Portanto, é preciso conceder explicitamente permissões a Bob para usar todas as ações do Amazon SQS, além da ação `CreateQueue` na política do IAM.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:*:123456789012:alice_queue_*"
   }]
}
```

Permitir que os desenvolvedores escrevam mensagens em uma fila compartilhada

No exemplo a seguir, criamos um grupo para desenvolvedores e anexamos uma política que permite que o grupo use a `SendMessage` ação do Amazon SQS, mas somente com a fila que pertence ao especificado Conta da AWS e é nomeada. `MyCompanyQueue`

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:*:123456789012:MyCompanyQueue"
   }]
}
```

Você pode usar * em vez de `SendMessage` para conceder as seguintes ações a um principal em uma fila compartilhada: `ChangeMessageVisibility`, `DeleteMessage`, `GetQueueAttributes`, `GetQueueUrl`, `ReceiveMessage` e `SendMessage`.

> **Note**
>
> Embora * inclua o acesso fornecido por outros tipos de permissão, o Amazon SQS considera as permissões separadamente. Por exemplo, é possível conceder permissões * e `SendMessage` a um usuário, embora * inclua o acesso fornecido pelo `SendMessage`. Esse conceito também se aplica quando você remove uma permissão. Se uma entidade principal tiver apenas uma permissão *, a solicitação de remoção de uma permissão `SendMessage` não deixará a entidade principal com uma permissão do tipo tudo, exceto. Em vez disso, a solicitação não fará nada, pois a entidade principal não tinha uma permissão `SendMessage` explícita. Para deixar a entidade principal apenas com a permissão `ReceiveMessage`, primeiro adicione a permissão `ReceiveMessage` e remova a permissão *.

Permita que os gerentes obtenham o tamanho geral das filas

No exemplo a seguir, criamos um grupo para gerentes e anexamos uma política que permite que o grupo use a `GetQueueAttributes` ação do Amazon SQS com todas as filas que pertencem à conta especificada. AWS

```
{
```

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

Permitir que um parceiro envie mensagens para uma fila específica

Você pode realizar essa tarefa usando uma política do Amazon SQS ou do IAM. Se seu parceiro tiver uma Conta da AWS, talvez seja mais fácil usar uma política do Amazon SQS. No entanto, qualquer usuário da empresa do parceiro que possua as credenciais AWS de segurança pode enviar mensagens para a fila. Para limitar o acesso a um determinado usuário ou aplicação, você deve tratar o parceiro como um usuário em sua própria empresa e usar uma política do IAM em vez de uma política do Amazon SQS.

Esse exemplo executa as seguintes ações:

1. Crie um grupo chamado WidgetCo para representar a empresa parceira.
2. Criar um usuário para o usuário ou aplicativo específico na empresa do parceiro que precisa de acesso.
3. Adicione o usuário ao grupo .
4. Associe uma política que ofereça ao grupo acesso apenas à ação `SendMessage` somente para a fila denominada `WidgetPartnerQueue`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
         "Effect": "Allow",
         "Action": "sqs:SendMessage",
         "Resource": "arn:aws:sqs:*:123456789012:WidgetPartnerQueue"
   }]
}
```

Exemplos básicos de políticas do Amazon SQS

Esta seção mostra exemplos de políticas para casos de uso comuns do Amazon SQS.

Você pode usar o console para verificar os efeitos de cada política à medida que anexa a política ao usuário. Inicialmente, o usuário não tem permissões e, portanto, não poderá fazer nada no console. À medida que você anexar políticas ao usuário, poderá verificar se o usuário pode executar várias ações no console.

> **Note**
>
> Recomendamos que você use duas janelas do navegador: uma para conceder permissões e outra para fazer login AWS Management Console usando as credenciais do usuário para verificar as permissões à medida que você as concede ao usuário.

Exemplo 1: Conceder uma permissão a um Conta da AWS

O exemplo de política a seguir Conta da AWS concede `111122223333` ao número a `SendMessage` permissão para a fila nomeada `444455556666/queue1` na região Leste dos EUA (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"
   }]
}
```

Exemplo 2: Conceder duas permissões a uma Conta da AWS

O exemplo de política a seguir Conta da AWS concede o número `111122223333` `SendMessage` e a `ReceiveMessage` permissão para a fila nomeada`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"
   }]
}
```

Exemplo 3: Conceder todas as permissões a dois Contas da AWS

O exemplo de política a seguir concede dois Contas da AWS números diferentes (111122223333e444455556666) permissão para usar todas as ações às quais o Amazon SQS permite acesso compartilhado para a fila nomeada `123456789012/queue1` na região Leste dos EUA (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"
   }]
}
```

Exemplo 4: conceder permissões entre contas a um perfil e um nome de usuário

O exemplo de política a seguir concede `role1` e, `username1` sob Conta da AWS número, permissão `111122223333` entre contas para usar todas as ações às quais o Amazon SQS permite acesso compartilhado à fila `123456789012/queue1` nomeada na região Leste dos EUA (Ohio).

As permissões entre contas não se aplicam às seguintes ações:

- `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"
   }]
```

```
}
```

Exemplo 5: conceder uma permissão a todos os usuários

O exemplo de política a seguir concede a todos os usuários (anônimos) a permissão `ReceiveMessage` para a fila 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"
   }]
}
```

Exemplo 6: conceder uma permissão de tempo limitado a todos os usuários

O exemplo de política a seguir concede a todos os usuários (anônimos) a permissão `ReceiveMessage` para a fila denominada `111122223333/queue1`, mas apenas das 12h (meio dia) às 15h em 31 de janeiro 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"
         }
      }
```

```
   }]
}
```

Exemplo 7: conceder todas as permissões a todos os usuários em um intervalo CIDR

A política de exemplo a seguir concede a todos os usuários (anônimos) permissão para usar todas as ações possíveis do Amazon SQS que podem ser compartilhadas para a fila denominada `111122223333/queue1`, mas somente se a solicitação vier do intervalo 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"
         }
      }
   }]
}
```

Exemplo 8: lista de permissões e lista de bloqueios para usuários em diferentes intervalos CIDR

O exemplo de política a seguir contém duas instruções:

- A primeira instrução concede a todos os usuários (anônimos) no intervalo `192.0.2.0/24` CIDR (exceto para `192.0.2.188`) permissão para usar a ação `SendMessage` para a fila denominada `111122223333`/queue1.
- A segunda instrução impede que todos os usuários (anônimos) no intervalo CIDR `12.148.72.0/23` usem a fila.

```
{
   "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"
         }
      }
   }]
}
```

Usar políticas personalizadas com linguagem de políticas de acesso do Amazon SQS

Se você quiser permitir o acesso ao Amazon SQS com base somente em um Conta da AWS ID e permissões básicas (como para `SendMessage`ou `ReceiveMessage`), você não precisa escrever suas próprias políticas. Basta usar a ação `AddPermission` do Amazon SQS.

Se você quiser negar ou permitir explicitamente o acesso com base em condições mais específicas (como a hora em que a solicitação chega ou o endereço IP do solicitante), você precisa escrever suas próprias políticas do Amazon SQS e carregá-las AWS no sistema usando a ação Amazon SQS. `SetQueueAttributes`

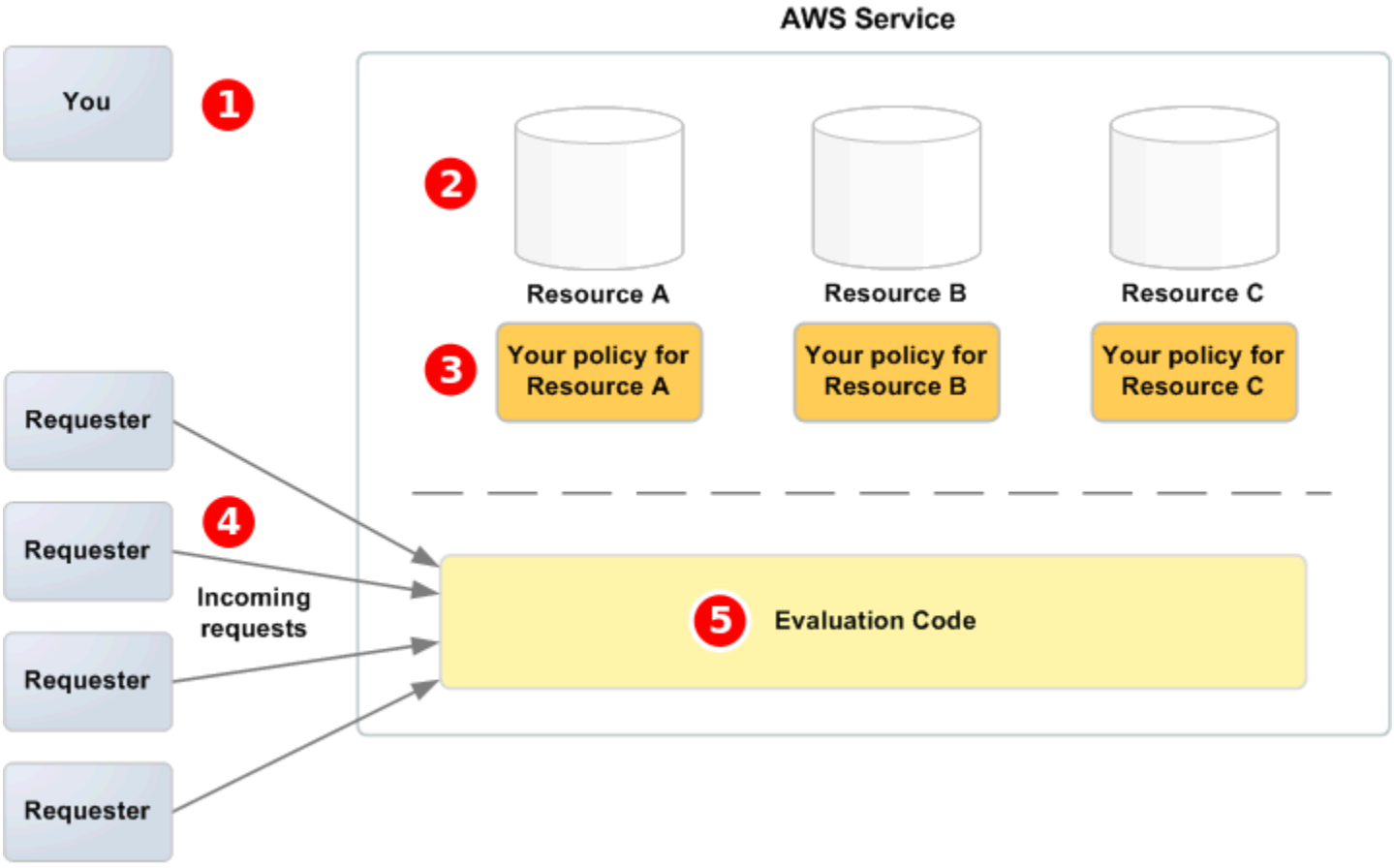
Tópicos

- Arquitetura do controle de acesso do Amazon SQS
- Fluxo de trabalho do processo de controle de acesso do Amazon SQS

- Conceitos-chave da linguagem de políticas de acesso do Amazon SQS
- Lógica de avaliação da linguagem de políticas de acesso do Amazon SQS
- Relações entre negações explícitas e padrão na linguagem de políticas de acesso do Amazon SQS
- Limitações das políticas personalizadas do Amazon SQS
- Exemplos de linguagem de políticas de acesso do Amazon SQS personalizadas

Arquitetura do controle de acesso do Amazon SQS

O diagrama a seguir descreve o controle de acesso para seus recursos do Amazon SQS.



1

Você, o proprietário do recurso.

2

Seus recursos contidos no AWS serviço (por exemplo, filas do Amazon SQS).

3

Suas políticas. É uma boa prática ter uma política por recurso. O AWS serviço fornece uma API que você usa para carregar e gerenciar suas políticas.

4

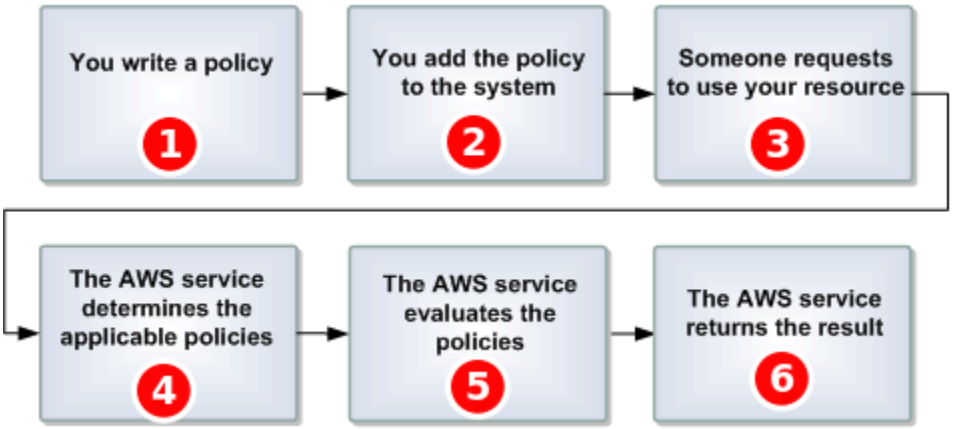
Os solicitantes e suas solicitações de entrada para o serviço da AWS .

5

O código de avaliação da linguagem de políticas de acesso. Esse é o conjunto de códigos dentro do AWS serviço que avalia as solicitações recebidas em relação às políticas aplicáveis e determina se o solicitante tem permissão para acessar o recurso.

Fluxo de trabalho do processo de controle de acesso do Amazon SQS

O diagrama a seguir descreve o fluxo de trabalho geral do controle de acesso com a linguagem de políticas de acesso do Amazon SQS.



1

Você cria uma política do Amazon SQS para a fila.

2

Você carrega sua política para AWS. O AWS serviço fornece uma API que você usa para carregar suas políticas. Por exemplo, você pode usar a ação `SetQueueAttributes` do Amazon SQS com a finalidade de fazer upload de uma política para uma determinada fila do Amazon SQS.

3

Alguém envia uma solicitação para usar sua fila do Amazon SQS.

4

O Amazon SQS examina todas as políticas do Amazon SQS disponíveis e determina quais são aplicáveis.

5

O Amazon SQS avalia as políticas e determina se o solicitante tem permissão para usar sua fila.

6

Com base no resultado da avaliação da política, o Amazon SQS retorna um erro `Access denied` para o solicitante ou continua a processar a solicitação.

Conceitos-chave da linguagem de políticas de acesso do Amazon SQS

Para criar suas próprias políticas, você deve estar familiarizado com JSON e um número de conceitos-chave.

Permitir

O resultado de uma Instrução que tenha Efeito definido como `allow`.

Ação

A atividade que o Principal tem permissão para executar, normalmente uma solicitação para à AWS.

Negação padrão

O resultado de uma Instrução que não tem as configurações Permitir e Negação explícita.

Condição

Qualquer restrição ou detalhe sobre uma Permissão. Condições comuns são relacionadas a data e hora e a endereços IP.

Efeito

O resultado que você deseja que a Instrução de uma Política retorno no momento da avaliação. Você especifica o valor `deny` ou `allow` ao gravar a declaração de política. Há três resultados possíveis no momento da avaliação da política: Negação padrão, Permitir e Negação explícita.

Negação explícita

O resultado de uma Instrução que tenha Efeito definido como `deny`.

Avaliação

O processo que o Amazon SQS usa para determinar se uma solicitação recebida deve ser negada ou permitida com base em uma Política.

Emissor

O usuário que grava uma Política para conceder permissões a um recurso. O emissor, por definição, é sempre o proprietário do recurso. AWS não permite que os usuários do Amazon SQS criem políticas para recursos que não possuem.

Chave

A característica específica que é a base para a restrição de acesso.

Permissão

O conceito de permissão ou não de acesso a um recurso usando uma Condição e uma Chave.

Política

O documento que atua como um contêiner para uma ou mais instruções.

O Amazon SQS usa a política para determinar se deverá conceder acesso a um usuário para um recurso.

Principal

O usuário que recebe Permissão na Política.

Recurso

O objeto ao qual a Principal solicita acesso.

Instrução

A descrição formal de uma única permissão, escrita na linguagem de políticas de acesso como parte de um documento de Política mais amplo.

Solicitante

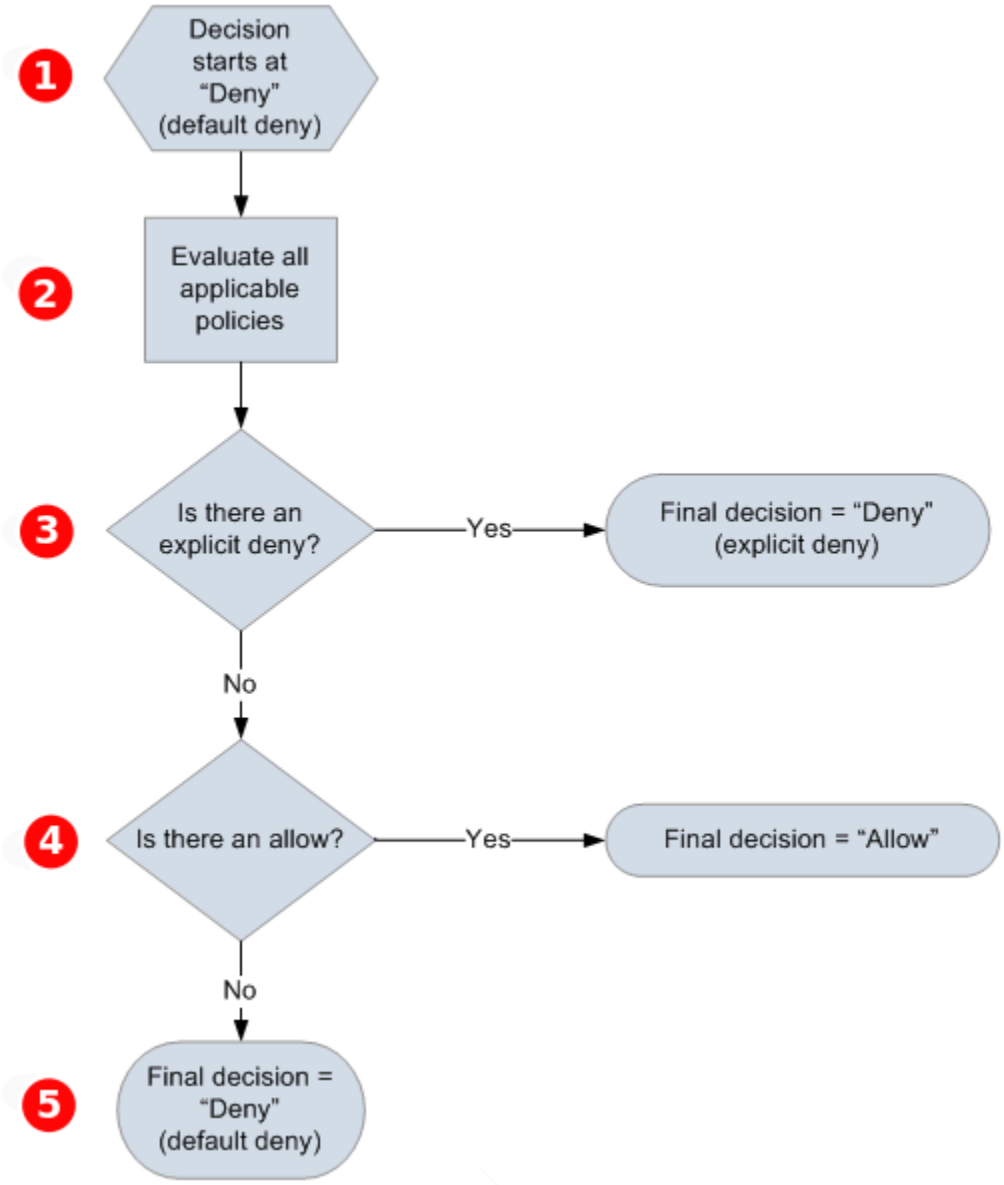
O usuário que envia uma solicitação de acesso a um Recurso.

Lógica de avaliação da linguagem de políticas de acesso do Amazon SQS

No momento da avaliação, o Amazon SQS determina se uma solicitação de alguém que não seja o proprietário do recurso deve ser permitida ou negada. A lógica de avaliação segue várias regras básicas:

- Por padrão, todas as solicitações para usar o recurso que venham de outras pessoas, e não de você, serão negadas.
- Um Permitir substitui qualquer Negação padrão.
- Uma Negação explícita substitui qualquer permissão.
- A ordem em que as políticas são avaliadas não é importante.

O diagrama a seguir descreve em detalhes como o Amazon SQS avalia as decisões sobre permissões de acesso.



1

A decisão começa com uma negação padrão.

2

O código de aplicação avalia todas as políticas que são aplicáveis à solicitação (com base no recurso, na entidade principal, na ação e nas condições). A ordem em que o código de aplicação avalia as políticas não é importante.

3

O código de imposição procura uma instrução de negação explícita que possa ser aplicada à

solicitação. Se encontrar um código, o código de aplicação retornará uma decisão de negação e o processo será concluído.

4

Se nenhuma instrução de negação explícita for encontrada, o código de imposição procurará qualquer instrução de permissão que possa ser aplicada à solicitação. Se encontrar uma instrução "permitir", o código de aplicação retornará uma decisão de permitir, e o processo será concluído (o serviço continua a processar a solicitação).

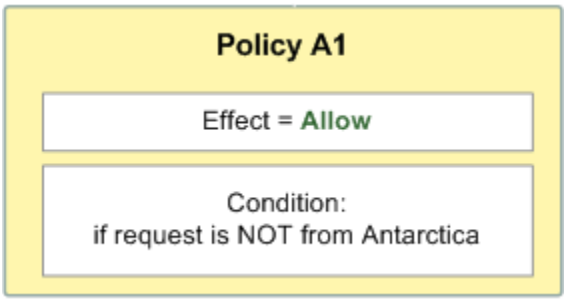
5

Se nenhuma instrução de permissão for encontrada, a decisão final será negar (como não há nenhuma negação explícita ou permissão, isso é considerado uma negação padrão).

Relações entre negações explícitas e padrão na linguagem de políticas de acesso do Amazon SQS

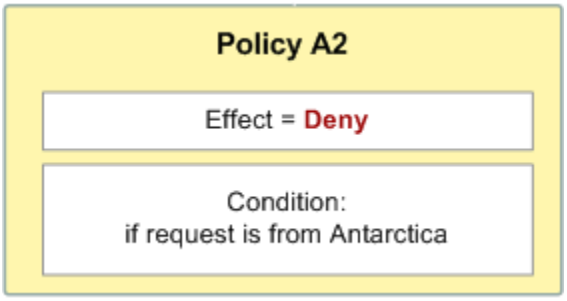
Se uma política do Amazon SQS não se aplicar diretamente a uma solicitação, esta resultará em uma Negação padrão. Por exemplo, se um usuário solicitar permissão para usar o Amazon SQS, mas a única política que se aplica a ele puder usar o DynamoDB, as solicitações resultarão em uma negação padrão.

Se uma condição em uma instrução não for atendida, a solicitação resultará em uma negação padrão. Se todas as condições em uma instrução forem atendidas, a solicitação resultará em uma Permitir ou uma Negação explícita, com base no valor do elemento Efeito da política. As políticas não especificam o que fazer se uma condição não for atendida, portanto, o resultado padrão nesse caso é uma negação padrão. Por exemplo, digamos que você deseje evitar solicitações da Antártica. Você elabora uma Política A1 que permite uma solicitação apenas se não vier da Antártica. O seguinte diagrama ilustra a política do Amazon SQS.



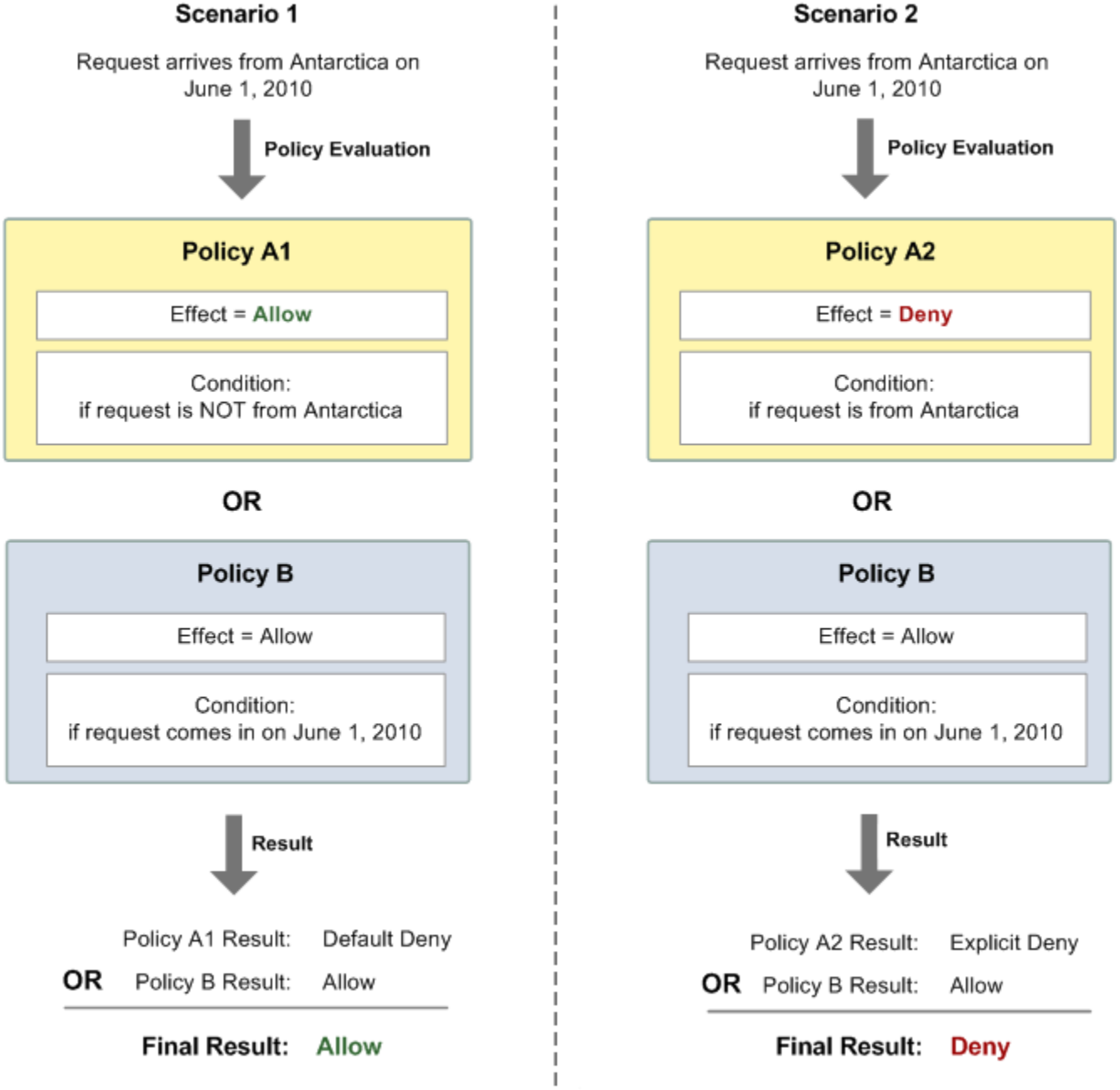
Se um usuário enviar uma solicitação dos EUA, a condição será atendida (a solicitação não será da Antártica) e a solicitação resultará em uma permissão. No entanto, se um usuário enviar uma solicitação da Antártica, a condição não será atendida, e a solicitação será padronizada para uma

solicitação padrão. Você pode alterar o resultado para uma negação explícita criando a Política A2 que nega explicitamente uma solicitação quando ela for recebida da Antártica. O seguinte diagrama ilustra a política.



Se um usuário enviar uma solicitação da Antártica, a condição será atendida, e a solicitação resultará em uma negação explícita.

A distinção entre uma negação padrão e uma negação explícita é importante porque uma permissão pode substituir a anterior, mas não a última. Por exemplo, a Política B permite solicitações que cheguem em 1º de junho de 2010. O diagrama a seguir compara a combinação dessa política com Política A1 e Política A2.



No Cenário 1, a Política A1 resulta em uma negação padrão e a Política B resulta em uma permissão porque a política permite solicitações recebidas em 1º de junho de 2010. A permissão da Política B substitui a negação padrão da Política A1, e a solicitação é permitida.

No Cenário 2, a Política B2 resulta em uma negação explícita, e a Política B resulta em uma permissão. A negação explícita da Política A2 substitui a permissão da Política B, e a solicitação é negada.

Limitações das políticas personalizadas do Amazon SQS

Acesso entre contas

As permissões entre contas não se aplicam às seguintes ações:

- `AddPermission`
- `CancelMessageMoveTask`
- `CreateQueue`
- `DeleteQueue`
- `ListMessageMoveTask`
- `ListQueues`
- `ListQueueTags`
- `RemovePermission`
- `SetQueueAttributes`
- `StartMessageMoveTask`
- `TagQueue`
- `UntagQueue`

Chaves de condição

Atualmente, o Amazon SQS suporta apenas um subconjunto limitado de chaves de condições disponíveis no IAM. Para ter mais informações, consulte Permissões da API do Amazon SQS: referência de ações e recurso.

Exemplos de linguagem de políticas de acesso do Amazon SQS personalizadas

Os seguintes são exemplos de políticas de acesso típicas do Amazon SQS.

Exemplo 1: conceder permissão a uma conta

O exemplo de política do Amazon SQS a seguir concede à Conta da AWS 111122223333 permissão para enviar e receber da `queue2`, de propriedade da Conta da 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"
   }]
}
```

Exemplo 2: conceder permissão a uma ou mais contas

O exemplo a seguir da política do Amazon SQS dá a um ou mais Contas da AWS acesso às filas pertencentes à sua conta por um período de tempo específico. É necessário criar essa política e fazer upload no Amazon SQS usando a ação `SetQueueAttributes`, porque a ação `AddPermission` não permite especificar uma restrição de tempo ao conceder acesso a uma fila.

```
{
   "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"
         }
      }
   }]
}
```

Exemplo 3: Conceder permissão para solicitações de instâncias do Amazon EC2

A política de exemplo do Amazon SQS a seguir fornece acesso a solicitações provenientes de instâncias do Amazon EC2. Esse exemplo se baseia no exemplo "Exemplo 2: conceder permissão a uma ou mais contas": ele restringe o acesso a antes de 30 de junho de 2009 ao meio-dia (UTC), que restringe o acesso ao intervalo de IP `203.0.113.0/24`. É necessário criar essa política e fazer upload no Amazon SQS usando a ação `SetQueueAttributes`, porque a ação `AddPermission` não permite especificar uma restrição de endereço IP ao conceder acesso a uma fila.

```
{
   "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"
         }
      }
   }]
}
```

Exemplo 4: negar acesso a uma conta específica

O exemplo a seguir da política do Amazon SQS nega um Conta da AWS acesso específico à sua fila. Este exemplo se baseia no exemplo Exemplo 1: conceder permissão a uma conta "": ele nega acesso ao especificado. Conta da AWSÉ necessário criar essa política e fazer upload no Amazon SQS usando a ação `SetQueueAttributes`, porque a ação `AddPermission` não permite negar acesso a uma fila (ela permite apenas conceder acesso a uma fila).

```
{
   "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"
   }]
}
```

Exemplo 5: negar o acesso se não vier de um VPC endpoint

O exemplo de política do Amazon SQS a seguir restringe o acesso à `queue1`: 111122223333 pode realizar as ações `SendMessage` e `ReceiveMessage` somente do ID de endpoint da VPC `vpce-1a2b3c4d` (especificado usando a condição `aws:sourceVpce`). Para ter mais informações, consulte Endpoints da Amazon Virtual Private Cloud para o Amazon SQS.

> **Note**
>
> - A condição `aws:sourceVpce` não requer um ARN para o recurso do VPC endpoint, somente o ID do VPC endpoint.
> - Você pode modificar o exemplo a seguir para restringir todas as ações para um endpoint da VPC negando todas as ações do Amazon SQS (`sqs:*`) na segunda instrução. No

entanto, essa instrução de política determinará que todas as ações (incluindo ações administrativas necessárias para modificar as permissões da fila) deverão ser feitas por meio do VPC endpoint específico definido na política, potencialmente impedindo que o usuário modifique as permissões da fila no 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"
            }
         }
      }
   ]
}
```

Usar credenciais de segurança temporárias com o Amazon SQS

Além de criar usuários com suas próprias credenciais de segurança, o IAM também permite que você conceda credenciais de segurança temporárias a qualquer usuário, permitindo que ele acesse seus AWS serviços e recursos. Você pode gerenciar os usuários que têm Contas da AWS. Você também pode gerenciar usuários do seu sistema que não têm Contas da AWS (usuários federados). Além disso, os aplicativos que você cria para acessar seus AWS recursos também podem ser considerados “usuários”.

Use essas credenciais de segurança temporárias para fazer solicitações ao Amazon SQS. As bibliotecas de API calculam o valor de assinatura necessário usando essas credenciais para autenticar sua solicitação. Se você enviar solicitações usando credenciais vencidas, o Amazon SQS negará a solicitação.

> **Note**
>
> Você não pode definir uma política com base em credenciais temporárias.

Pré-requisitos

1. Use o IAM para criar credenciais de segurança temporárias:
   - Token de segurança
   - Access Key ID
   - Secret Access Key
2. Prepare sua string para assinar com o ID da chave de acesso temporária e o token de segurança.
3. Use a chave de acesso secreta temporária em vez de sua própria chave de acesso secreta para assinar a solicitação de API de consulta.

> **Note**
>
> Quando você enviar a solicitação de API de consulta assinada, use o ID da chave de acesso temporária em vez de seu próprio ID da chave de acesso e para incluir o token de segurança. Para obter mais informações sobre o suporte do IAM para credenciais de

segurança temporárias, consulte Como conceder acesso temporário aos seus AWS recursos no Guia do usuário do IAM.

Para chamar uma ação de API de consulta do Amazon SQS usando credenciais de segurança temporárias

1. Solicite um token de segurança temporário usando AWS Identity and Access Management o. Para obter mais informações, consulte Criar credencias de segurança temporárias para habilitar o acesso para usuários do IAM no Guia do usuário do IAM.

   O IAM retorna um token de segurança, um ID da chave de acesso e uma chave de acesso secreta.
2. Prepare sua consulta usando o ID da chave de acesso temporária em vez de seu próprio ID da chave de acesso e para incluir o token de segurança. Assine sua solicitação usando a chave de acesso secreta temporária em vez de sua própria.
3. Envie sua string de consulta assinada com o ID da chave de acesso temporária e o token de segurança.

   O exemplo a seguir demonstra como usar credenciais de segurança temporárias para autenticar uma solicitação do Amazon SQS. A estrutura de *AUTHPARAMS* depende de como você assina sua solicitação de API. Para obter mais informações, consulte Assinar solicitações de AWS API na Referência geral da 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
   ```

   O exemplo a seguir usa credenciais de segurança temporárias para enviar duas mensagens usando a ação `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
```

Gerenciamento de acesso para filas criptografadas do Amazon SQS com políticas de privilégios mínimos

É possível usar o Amazon SQS para trocar dados sigilosos entre aplicações usando a criptografia do lado do servidor (SSE) integrada ao AWS Key Management Service (KMS). Com a integração do Amazon SQS e AWS KMS, você pode gerenciar centralmente as chaves que protegem o Amazon SQS, bem como as chaves que protegem seus outros recursos. AWS

Vários AWS serviços podem atuar como fontes de eventos que enviam eventos para o Amazon SQS. Para permitir que uma fonte de eventos acesse a fila criptografada do Amazon SQS, você precisa configurar a fila com uma chave gerenciada pelo cliente. AWS KMS Em seguida, use a política de chaves para permitir que o serviço use os métodos de AWS KMS API necessários. O serviço também exige permissões para autenticar o acesso e permitir que a fila envie eventos. É possível fazer isso usando uma política do Amazon SQS, que é uma política baseada em recursos que você pode usar para controlar o acesso à fila do Amazon SQS e os respectivos dados.

As seções a seguir fornecem informações sobre como controlar o acesso à sua fila criptografada do Amazon SQS por meio da política do Amazon SQS e da política de chaves. AWS KMS As políticas deste guia ajudarão você a alcançar o privilégio mínimo.

Este guia também descreve como as políticas baseadas em recursos resolvem o problema de adjuntos confundidos usando as chaves de contexto de condição globais do IAM `aws:SourceArn`, `aws:SourceAccount` e `aws:PrincipalOrgID`.

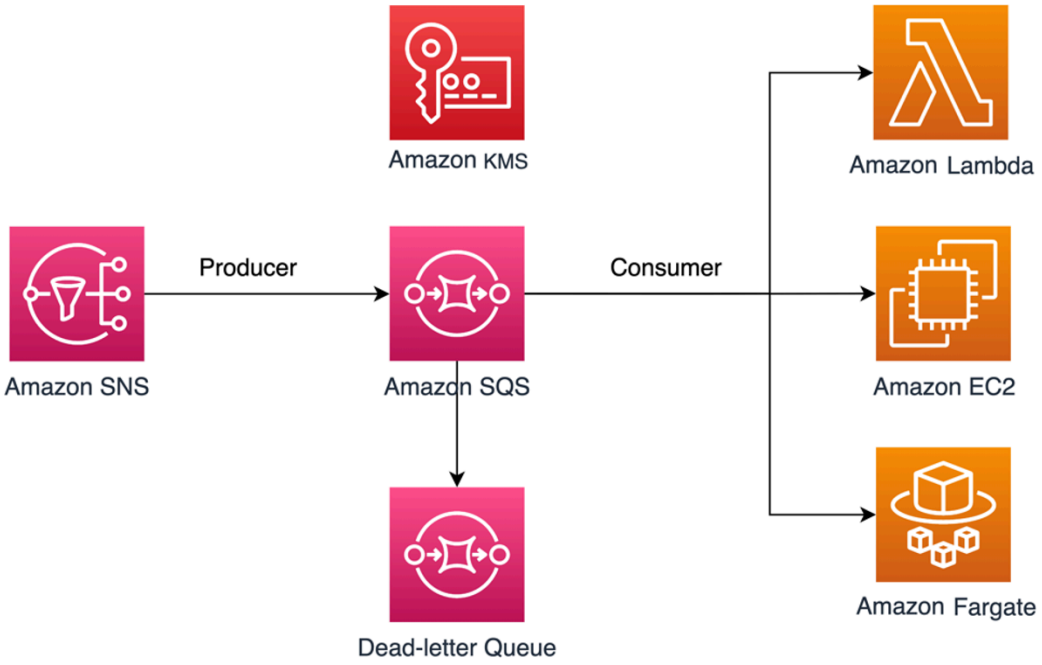
Tópicos

- Visão geral

- Política de chaves de privilégio mínimo para o Amazon SQS
- Declarações de política do Amazon SQS para a fila de mensagens não entregues
- Prevenção do problema de adjunto confuso entre serviços
- Usar o IAM Access Analyzer para analisar o acesso entre contas

Visão geral

Neste tópico, vamos mostrar um caso de uso comum para ilustrar como você pode criar a política de chaves e a política de filas do Amazon SQS. Esse caso de uso é mostrado na imagem a seguir.



Neste exemplo, o produtor de mensagens é um tópico do Amazon Simple Notification Service (SNS), que está configurado para fazer fanout de mensagens para a fila criptografada do Amazon SQS. O consumidor de mensagens é um serviço de computação, como uma função do AWS Lambda, uma instância do Amazon Elastic Compute Cloud (EC2) ou um contêiner do AWS Fargate. Sua fila do Amazon SQS é então configurada para enviar mensagens de falha Amazon a uma fila de mensagens não entregues (DLQ). Isso é útil para depurar a aplicação ou o sistema de mensagens, pois as DLQs permitem que você isole mensagens não consumidas para determinar por que seu processamento não obteve êxito. Na solução definida neste tópico, um serviço de computação, como uma função do Lambda, é usado para processar mensagens armazenadas na fila do Amazon SQS. Se o consumidor da mensagem estiver localizado em uma nuvem privada virtual (VPC), a declaração de política `DenyReceivingIfNotThroughVPCE` incluída neste guia permite restringir o recebimento de mensagens a essa VPC específica.

> **Note**
>
> Este guia contém somente as permissões do IAM necessárias na forma de declarações de política. Para criar a política, você precisa adicionar as declarações à sua política do Amazon SQS ou à sua política de AWS KMS chaves. Este guia não fornece instruções sobre como criar a fila ou a chave do Amazon SQS. AWS KMS Para obter instruções sobre a criação desses recursos, consulte "Creating an Amazon SQS queue" (Criar uma fila do Amazon SQS) e Creating keys (Criar chaves).
> A política do Amazon SQS definida neste guia não é compatível com o redirecionamento de mensagens diretamente para a mesma fila ou uma fila diferente do Amazon SQS.

Política de chaves de privilégio mínimo para o Amazon SQS

Nesta seção, descrevemos as permissões de privilégio mínimo necessárias AWS KMS para a chave gerenciada pelo cliente que você usa para criptografar sua fila do Amazon SQS. Com essas permissões, você pode limitar o acesso somente às entidades pretendidas e, ao mesmo tempo, implementar o privilégio mínimo. A política principal deve consistir nas seguintes declarações de política, que descrevemos em detalhes abaixo:

- Conceda permissões de administrador à AWS KMS chave
- Conceder acesso somente leitura aos metadados de chave
- Conceder permissões de KMS do Amazon SNS para que este publique mensagens na fila
- Permitir que os consumidores decodifiquem mensagens da fila

Conceda permissões de administrador à AWS KMS chave

Para criar uma AWS KMS chave, você precisa fornecer permissões de AWS KMS administrador para a função do IAM que você usa para implantar a AWS KMS chave. Essas permissões de administrador são definidas na declaração de política do `AllowKeyAdminPermissions` a seguir. Ao adicionar essa declaração à sua política de AWS KMS chaves, certifique-se de <admin-role ARN>substituí-la pelo Amazon Resource Name (ARN) da função do IAM usada para implantar a AWS KMS chave, gerenciar a AWS KMS chave ou ambas. Essa pode ser o perfil do IAM do pipeline de implantação ou a função de administrador da sua organização no 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**
>
> Em uma política AWS KMS chave, o valor do `Resource` elemento precisa ser`*`, o que significa “essa AWS KMS chave”. O asterisco (`*`) identifica a AWS KMS chave à qual a política de chaves está anexada.

Conceder acesso somente leitura aos metadados de chave

Para conceder a outros perfis do IAM acesso somente leitura aos metadados de chave, adicione a declaração `AllowReadAccessToKeyMetaData` à política de chaves. Por exemplo, a declaração a seguir permite que você liste todas as AWS KMS chaves em sua conta para fins de auditoria. Essa declaração concede ao usuário AWS raiz acesso somente de leitura aos metadados da chave. Portanto, qualquer entidade principal do IAM na conta pode ter acesso aos metadados de chave quando as respectivas políticas baseadas em identidade tiverem as permissões listadas na seguinte declaração: `kms:Describe*`, `kms:Get*` e `kms:List*`. Substitua *`<account-ID>`* por suas próprias informações.

```
{
  "Sid": "AllowReadAcesssToKeyMetaData",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::<accountID>:root"
    ]
  },
  "Action": [
    "kms:Describe*",
    "kms:Get*",
    "kms:List*"
  ],
  "Resource": "*"
}
```

Conceder permissões de KMS do Amazon SNS para que este publique mensagens na fila

Para permitir que o tópico do Amazon SNS publique mensagens na fila criptografada do Amazon SQS, adicione a declaração de política `AllowSNSToSendToSQS` à sua política de chaves. Essa declaração concede ao Amazon SNS permissões para usar a AWS KMS chave para publicar na sua fila do Amazon SQS. Substitua *`<account-ID>`* por suas próprias informações.

> **Note**
>
> O `Condition` código na declaração limita o acesso somente ao serviço Amazon SNS na mesma AWS conta.

```
{
  "Sid": "AllowSNSToSendToSQS",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "sns.amazonaws.com"
    ]
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
```

```
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "<account-id>"
    }
  }
}
```

Permitir que os consumidores decodifiquem mensagens da fila

A declaração `AllowConsumersToReceiveFromTheQueue` a seguir concede ao consumidor de mensagens do Amazon SQS as permissões necessárias para descriptografar as mensagens recebidas da fila criptografada do Amazon SQS. Ao anexar a declaração de política, substitua *<consumer's runtime role ARN>* pelo ARN da função de tempo de execução do IAM do consumidor da mensagem.

```
{
  "Sid": "AllowConsumersToReceiveFromTheQueue",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "<consumer's execution role ARN>"
    ]
  },
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Política de privilégio mínimo do Amazon SQS

Esta seção mostra as políticas de fila do Amazon SQS de privilégio mínimo para o caso de uso coberto por este guia (por exemplo, do Amazon SNS para o Amazon SQS). A política definida foi projetada para impedir o acesso não intencional usando uma combinação das declarações `Deny` e `Allow`. As declarações `Allow` concedem acesso à entidade ou às entidades pretendidas. As declarações `Deny` impedem que outras entidades não intencionais acessem a fila do Amazon SQS e exclui a entidade pretendida da condição da política.

A política do Amazon SQS inclui as seguintes declarações, que descrevemos em detalhes abaixo:

- Restringir as permissões de gerenciamento do Amazon SQS

- Restringir as ações de fila do Amazon SQS da organização especificada
- Conceder permissões do Amazon SQS aos consumidores
- Aplicar a criptografia em trânsito
- Restringir a transmissão de mensagens para um tópico específico do Amazon SNS
- (Opcional) Restringir o recebimento de mensagens a um endpoint da VPC específico

Restringir as permissões de gerenciamento do Amazon SQS

A declaração de política `RestrictAdminQueueActions` a seguir restringe as permissões de gerenciamento do Amazon SQS somente ao perfil ou aos perfis do IAM que você usa para implantar a fila, gerenciar a fila ou realizar ambas as atividades. Substitua *<placeholder values>* por suas próprias informações. Especifique o ARN do perfil do IAM usado para implantar a fila do Amazon SQS, bem como os ARNs de qualquer função de administrador que deva ter permissões de gerenciamento do 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>"
      ]
    }
  }
}
```

Restringir as ações de fila do Amazon SQS da organização especificada

Para ajudar a proteger seus recursos do Amazon SQS contra acesso externo (acesso por uma entidade fora da sua organização da AWS), use a instrução a seguir. Essa declaração limita o acesso à fila do Amazon SQS à organização que você especifica na `Condition`. Substitua *<SQS queue ARN>* pelo ARN do perfil do IAM usado para implantar a fila do Amazon SQS, e *<org-id>*, pelo ID da organização.

```
{
  "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>"
      ]
    }
  }
}
```

Conceder permissões do Amazon SQS aos consumidores

Para receber mensagens da fila do Amazon SQS, você precisa fornecer ao consumidor da mensagem as permissões necessárias. A declaração de política a seguir concede ao consumidor, especificado por você, as permissões necessárias para consumir mensagens da fila do Amazon SQS. Ao adicionar a declaração à política do Amazon SQS, substitua *<consumer's IAM runtime role ARN>* pelo ARN da função de tempo de execução do IAM usada pelo consumidor, e *<SQS queue ARN>* pelo ARN do perfil do IAM usado para implantar a fila do 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 evitar que outras entidades recebam mensagens da fila do Amazon SQS, adicione a declaração `DenyOtherConsumersFromReceiving` à política de filas do Amazon SQS. Essa declaração restringe o consumo de mensagens ao consumidor que você especificar, permitindo que nenhum outro consumidor tenha acesso, mesmo quando suas permissões de identidade concederem acesso. Substitua *<SQS queue ARN>* e *<consumer's runtime role ARN>* por suas próprias informações.

```
{
  "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>"
    }
  }
}
```

Aplicar a criptografia em trânsito

A declaração de política do `DenyUnsecureTransport` a seguir obriga os consumidores e produtores a usarem canais seguros (conexões TLS) para enviar e receber mensagens da fila do Amazon SQS. Substitua *<SQS queue ARN>* pelo ARN do perfil do IAM usado para implantar a fila do Amazon SQS.

```
{
  "Sid": "DenyUnsecureTransport",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": [
    "sqs:ReceiveMessage",
    "sqs:SendMessage"
  ],
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "Bool": {
      "aws:SecureTransport": "false"
    }
  }
}
```

Restringir a transmissão de mensagens para um tópico específico do Amazon SNS

Veja a seguir um exemplo de declaração de política que permite ao tópico do Amazon SNS enviar mensagem à fila do Amazon SQS. Substitua *<SQS queue ARN>* pelo ARN do perfil do IAM usado para implantar a fila do Amazon SQS, e *<SNS topic ARN>* pelo ARN do tópico do 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>"
    }
  }
}
```

A declaração de política `DenyAllProducersExceptSNSFromSending` a seguir impede que outros produtores enviem mensagens à fila. Substitua *<SQS] queue ARN>* e *<SNS topic ARN>* por suas próprias informações.

```
{
  "Sid": "DenyAllProducersExceptSNSFromSending",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
  "Action": "sqs:SendMessage",
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "ArnNotLike": {
      "aws:SourceArn": "<SNS topic ARN>"
    }
  }
}
```

(Opcional) Restringir o recebimento de mensagens a um endpoint da VPC específico

Para restringir o recebimento de mensagens apenas a determinado endpoint da VPC, adicione a declaração de política do Amazon SQS à política de fila do Amazon SQS. Essa declaração impede que um consumidor de mensagens receba mensagens da fila, a menos que as mensagens sejam do endpoint da VPC desejado. Substitua *<SQS queue ARN>* pelo ARN do perfil do IAM usado para implantar a fila do Amazon SQS; e *<vpce_id>* pelo ID do endpoint da VPC.

```
{
  "Sid": "DenyReceivingIfNotThroughVPCE",
  "Effect": "Deny",
  "Principal": "*",
  "Action": [
    "sqs:ReceiveMessage"
```

```
  ],
  "Resource": "<SQS queue ARN>",
  "Condition": {
    "StringNotEquals": {
      "aws:sourceVpce": "<vpce id>"
    }
  }
}
```

Declarações de política do Amazon SQS para a fila de mensagens não entregues

Adicione as seguintes declarações de política, identificadas pelo ID da declaração, à política de acesso da fila de mensagens não entregues (DLQ):

- `RestrictAdminQueueActions`
- `DenyQueueActionsOutsideOrg`
- `AllowConsumersToReceiveFromTheQueue`
- `DenyOtherConsumersFromReceiving`
- `DenyUnsecureTransport`

Além de adicionar as declarações de política anteriores à política de acesso da DLQ, você também deve adicionar uma declaração para restringir a transmissão de mensagens às filas do Amazon SQS, conforme descrito na seção a seguir.

Restringir a transmissão de mensagens para filas do Amazon SQS

Para restringir o acesso somente às filas do Amazon SQS da mesma conta, adicione a declaração de política `DenyAnyProducersExceptSQS` a seguir à política de DLQs. Essa declaração não limita a transmissão de mensagens para uma fila específica porque é necessário implantar a DLQ antes de criar a fila principal. Por isso, você não saberá o ARN do Amazon SQS ao criar a DLQ. Se você precisar limitar o acesso a apenas uma fila do Amazon SQS, modifique `aws:SourceArn` na `Condition` com o ARN da sua fila de origem do Amazon SQS quando souber.

```
{
  "Sid": "DenyAnyProducersExceptSQS",
  "Effect": "Deny",
  "Principal": {
    "AWS": "*"
  },
```

```
  "Action": "sqs:SendMessage",
  "Resource": "<SQS DLQ ARN>",
  "Condition": {
    "ArnNotLike": {
      "aws:SourceArn": "arn:aws:sqs:<region>:<account-id>:*"
    }
  }
}
```

> **Important**
>
> As políticas de fila do Amazon SQS definidas neste guia não restringem à ação `sqs:PurgeQueue` para determinado perfil ou perfis do IAM. A ação `sqs:PurgeQueue` permite que você exclua todas as mensagens na fila do Amazon SQS. Também é possível usar essa ação para fazer alterações no formato da mensagem sem substituir a fila do Amazon SQS. Ao depurar uma aplicação, você pode limpar a fila do Amazon SQS para remover mensagens possivelmente errôneas. Ao testar a aplicação, você pode direcionar um alto volume de mensagens pela fila do Amazon SQS e, depois, limpar a fila para começar do zero antes de entrar em produção. O motivo para não restringir essa ação a determinada função é que essa função pode não ser conhecida ao implantar a fila do Amazon SQS. Você precisará adicionar essa permissão à política baseada em identidades da função para poder limpar a fila.

Prevenção do problema de adjunto confuso entre serviços

O problema de adjunto confuso é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir outra entidade mais privilegiada a executá-la. Para evitar isso, AWS fornece ferramentas que ajudam você a proteger sua conta se você fornecer acesso a terceiros (conhecido como contas cruzadas) ou outros AWS serviços (conhecido como serviços cruzados) aos recursos em sua conta. As declarações de política nesta seção podem ajudar a evitar o problema de adjunto confuso entre serviços.

A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado para que ele use as respectivas permissões com o objetivo de acessar os recursos de outro cliente de uma forma que, normalmente, ele não deveria ter permissão. Para ajudar combater esse problema, as políticas baseadas em recursos definidas nesta publicação usam as chaves de contexto de condição globais do IAM `aws:SourceArn`, `aws:SourceAccount` e `aws:PrincipalOrgID`. Isso

limita as permissões que um serviço tem para um recurso específico, uma conta específica ou uma organização específica em AWS Organizations.

Usar o IAM Access Analyzer para analisar o acesso entre contas

Você pode usar o AWS IAM Access Analyzer para revisar suas políticas de filas AWS KMS e políticas de chaves do Amazon SQS e alertá-lo quando uma fila do Amazon SQS ou AWS KMS uma chave concede acesso a uma entidade externa. O IAM Access Analyzer ajuda a identificar os recursos da sua organização e as contas que são compartilhadas com uma entidade fora da zona de confiança. Essa zona de confiança pode ser uma AWS conta ou a organização dentro de AWS Organizations que você especifica ao habilitar o IAM Access Analyzer.

O IAM Access Analyzer identifica recursos compartilhados com diretores externos usando o raciocínio baseado em lógica para analisar as políticas baseadas em recursos em seu ambiente. AWS Para cada instância de um recurso compartilhado fora de sua zona de confiança, o Access Analyzer gera uma descoberta. As descobertas incluem informações sobre o acesso e a entidade principal externa a que é concedido. Analise as descobertas para determinar se o acesso é pretendido e seguro ou se não é intencional e representa um risco à segurança. Para qualquer acesso não intencional, avalie a política afetada e corrija-a. Consulte esta postagem do blog para obter mais informações sobre como o AWS IAM Access Analyzer identifica o acesso não intencional aos seus recursos. AWS

Para obter mais informações sobre o AWS IAM Access Analyzer, consulte a documentação do AWS IAM Access Analyzer.

Permissões da API do Amazon SQS: referência de ações e recurso

Ao configurar o Controle de acesso e criar políticas de permissões que você possa anexar a uma identidade do IAM, é possível usar a tabela a seguir como referência. A inclui cada ação do Amazon Simple Queue Service, as ações correspondentes para as quais você pode conceder permissões para realizar a ação e o AWS recurso para o qual você pode conceder as permissões.

Especifique as ações no campo `Action` da política e o valor do recurso no campo `Resource` da política. Para especificar uma ação, use o prefixo `sqs:` seguido do nome da ação (por exemplo, `sqs:CreateQueue`).

No momento, o Amazon SQS é compatível apenas com chaves de contexto de condições globais disponíveis no IAM.

API do Amazon Simple Queue Service e permissões necessárias para ações

AddPermission

Ação/Ações: `sqs:AddPermission`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

ChangeMessageVisibilidade

Ação/Ações: `sqs:ChangeMessageVisibility`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

ChangeMessageVisibilityBatch

Ação/Ações: `sqs:ChangeMessageVisibilityBatch`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

CreateQueue

Ação/Ações: `sqs:CreateQueue`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

DeleteMessage

Ação/Ações: `sqs:DeleteMessage`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

DeleteMessageBatch

Ação/Ações: `sqs:DeleteMessageBatch`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

DeleteQueue

Ação/Ações: `sqs:DeleteQueue`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

GetQueueAtributos

Ação/Ações: `sqs:GetQueueAttributes`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

GetQueueURL

Ação/Ações: `sqs:GetQueueUrl`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

ListDeadLetterSourceFilas

Ação/Ações: `sqs:ListDeadLetterSourceQueues`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

ListQueues

Ação/Ações: `sqs:ListQueues`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

ListQueueEtiquetas

Ação/Ações: `sqs:ListQueueTags`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

PurgeQueue

Ação/Ações: `sqs:PurgeQueue`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

ReceiveMessage

Ação/Ações: `sqs:ReceiveMessage`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

RemovePermission

Ação/Ações: `sqs:RemovePermission`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

SendMessagee SendMessageBatch

Ação/Ações: `sqs:SendMessage`

Recurso: `arn:aws:sqs:`*`region`*`:`*`account_id`*`:`*`queue_name`*

SetQueueAtributos

Ação/Ações: `sqs:SetQueueAttributes`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

TagQueue

Ação/Ações: `sqs:TagQueue`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

UntagQueue

Ação/Ações: `sqs:UntagQueue`

Recurso: `arn:aws:sqs:region:account_id:queue_name`

Registrar em log e monitorar no Amazon SQS

Esta seção fornece informações sobre as opções de registro e monitoramento do Amazon SQS, incluindo como usar para capturar chamadas de API e CloudWatch métricas CloudTrail para obter informações sobre a atividade e o desempenho das filas.

Tópicos

- Registrar em log chamadas de API do Amazon SQS usando o AWS CloudTrail
- Monitoramento de filas do Amazon SQS usando CloudWatch

Registrar em log chamadas de API do Amazon SQS usando o AWS CloudTrail

O Amazon SQS é integrado AWS CloudTrail para registrar as chamadas do Amazon SQS de um usuário, função ou serviço. AWS CloudTrail captura chamadas de API relacionadas ao padrão do Amazon SQS e às filas FIFO como eventos, incluindo interações iniciadas por meio do console do Amazon SQS e programaticamente por meio de chamadas para as APIs do Amazon SQS.

Tópicos

- Informações do Amazon SQS em CloudTrail
- Eventos de gestão em CloudTrail
- Eventos de dados em CloudTrail
- Exemplos: eventos CloudTrail de gerenciamento para o Amazon SQS
- Exemplos: eventos CloudTrail de dados para o Amazon SQS

Informações do Amazon SQS em CloudTrail

CloudTrail é ativado por padrão quando você cria sua AWS conta. Quando ocorre uma atividade de evento compatível do Amazon SQS, ela é registrada em um CloudTrail evento, junto com outros eventos de AWS serviço, no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes para sua AWS conta. Para obter mais informações, consulte Visualização de CloudTrail eventos com histórico de eventos no Guia AWS CloudTrail do usuário.

As APIs do Amazon SQS que chamam operações de gerenciamento de filas, como, `AddPermission` são categorizadas como eventos de gerenciamento e são registradas por padrão. CloudTrail As APIs do Amazon SQS que são operações de alto volume executadas em uma fila do Amazon SQS, como `SendMessage` são categorizadas como eventos de dados e são registradas depois que você se inscreve. CloudTrail

Usando as informações CloudTrail coletadas, você pode identificar uma solicitação específica para uma API do Amazon SQS, o endereço IP ou a identidade do solicitante e a data e a hora da solicitação. Se você configurar uma CloudTrail trilha, poderá entregar CloudTrail eventos continuamente em um bucket do Amazon S3 com uma entrega opcional para o Amazon CloudWatch Logs e. AWS EventBridge Se você não configurar uma trilha, só poderá visualizar o histórico de eventos de gerenciamento em eventos no CloudTrail console. Para mais informações, consulte Visão geral da criação de uma trilha no Guia do usuário do AWS CloudTrail.

Eventos de gestão em CloudTrail

O Amazon SQS registra em log as seguintes ações de API como eventos de gerenciamento:

- `AddPermission`
- `CreateQueue`
- `CancelMessageMoveTask`
- `DeleteQueue`
- `ListMessageMoveTasks`
- `PurgeQueue`
- `RemovePermission`
- `SetQueueAttributes`
- `StartMessageMoveTask`
- `TagQueue`

- `UntagQueue`

As seguintes APIs do Amazon SQS não são suportadas para registro em log: CloudTrail

- `GetQueueAttributes`
- `GetQueueUrl`
- `ListDeadLetterSourceQueues`
- `ListQueueTags`
- `ListQueues`

Eventos de dados em CloudTrail

Eventos de dados fornecem informações sobre as operações de recursos realizadas em um recurso, como enviar ou receber uma mensagem do Amazon SQS de e para uma fila do Amazon SQS. Eventos de dados são atividades de alto volume que CloudTrail não são registradas por padrão. Você pode ativar o registro de ações da API de eventos de dados para sua fila do SQS usando CloudTrail APIs. Para obter mais informações, consulte Registrar eventos de dados, no Guia do usuário do AWS CloudTrail .

Com CloudTrail, você pode usar seletores de eventos avançados para decidir quais atividades da API do Amazon SQS são registradas e registradas. Para registrar em log eventos de dados do Amazon SQS, você deve incluir o tipo de recurso `AWS::SQS::Queue`. Depois de configurado, é possível refinar ainda mais suas preferências de registro em log selecionando eventos de dados específicos para gravação, como usar o filtro `eventName` para rastrear eventos `SendMessage`. Para obter mais informações, consulte `AdvancedEventSelector` na Referência de APIs do AWS CloudTrail .

Eventos de dados do Amazon SQS:

- `SendMessage`
- `SendMessageBatch`
- `ReceiveMessage`
- `DeleteMessage`
- `DeleteMessageBatch`
- `ChangeMessageVisibility`

- `ChangeMessageVisibilityBatch`

Há cobranças adicionais para eventos de dados. Para obter mais informações, consulte Preços do AWS CloudTrail.

Exemplos: eventos CloudTrail de gerenciamento para o Amazon SQS

Os exemplos a seguir mostram entradas de CloudTrail registro para APIs compatíveis:

AddPermission

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `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

O exemplo a seguir mostra uma 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"
    }
  ]
}
```

Exemplos: eventos CloudTrail de dados para o Amazon SQS

A seguir estão exemplos de CloudTrail eventos específicos das APIs de eventos de dados do Amazon SQS:

SendMessage

O exemplo a seguir mostra um evento CloudTrail de dados 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

O exemplo a seguir mostra um evento CloudTrail de dados 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

O exemplo a seguir mostra um evento CloudTrail de dados 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

O exemplo a seguir mostra um evento CloudTrail de dados 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"
  }
}
```

Monitoramento de filas do Amazon SQS usando CloudWatch

O Amazon SQS e o Amazon CloudWatch são integrados para que você possa usar CloudWatch para visualizar e analisar métricas para suas filas do Amazon SQS. Você pode visualizar e analisar as métricas de suas filas no console do Amazon SQS, no console, usando CloudWatch o ou usando AWS CLIa API. CloudWatch Você também pode definir CloudWatch alarmes para as métricas do Amazon SQS.

CloudWatch métricas para suas filas do Amazon SQS são coletadas e enviadas automaticamente CloudWatch em intervalos de um minuto. Essas métricas são coletadas em todas as filas que atendem às CloudWatch diretrizes para serem ativas. CloudWatchconsidera uma fila ativa por até seis horas se ela contiver alguma mensagem ou se alguma ação a acessar.

Quando uma fila do Amazon SQS fica inativa por mais de seis horas, o serviço Amazon SQS é considerado inativo e deixa de fornecer métricas para o serviço. CloudWatch Dados ausentes, ou dados representando zero, não podem ser visualizados nas CloudWatch métricas do Amazon SQS durante o período em que sua fila do Amazon SQS estava inativa.

> **Note**
>
> - Uma fila do Amazon SQS pode ser ativada quando o usuário que está chamando uma API na fila não está autorizado e a solicitação falha.
> - O console do Amazon SQS executa uma chamada de `GetQueueAttributes` API quando a página da fila é aberta. A solicitação `GetQueueAttributes` da API ativa a fila.
> - Um atraso de até 15 minutos ocorre nas CloudWatch métricas quando uma fila é ativada a partir de um estado inativo.

- Não há cobrança pelas métricas do Amazon SQS relatadas em. CloudWatch Elas são fornecidas como parte do serviço Amazon SQS.
- CloudWatch as métricas são compatíveis com filas padrão e FIFO.

Tópicos

Acessando CloudWatch métricas para o Amazon SQS

O Amazon SQS e o Amazon CloudWatch são integrados para que você possa usar CloudWatch para visualizar e analisar métricas para suas filas do Amazon SQS. Você pode visualizar e analisar as métricas de suas filas no console do Amazon SQS, no console, usando CloudWatch o ou usando AWS CLIa API. CloudWatch Você também pode definir CloudWatch alarmes para as métricas do Amazon SQS.

Console do Amazon SQS

1. Faça login no console do Amazon SQS.
2. Na lista de filas, escolha (selecione) as caixas das filas cujas métricas você deseja acessar. Você pode exibir métricas para até 10 filas.
3. Escolha a guia Monitoring (Monitoramento).

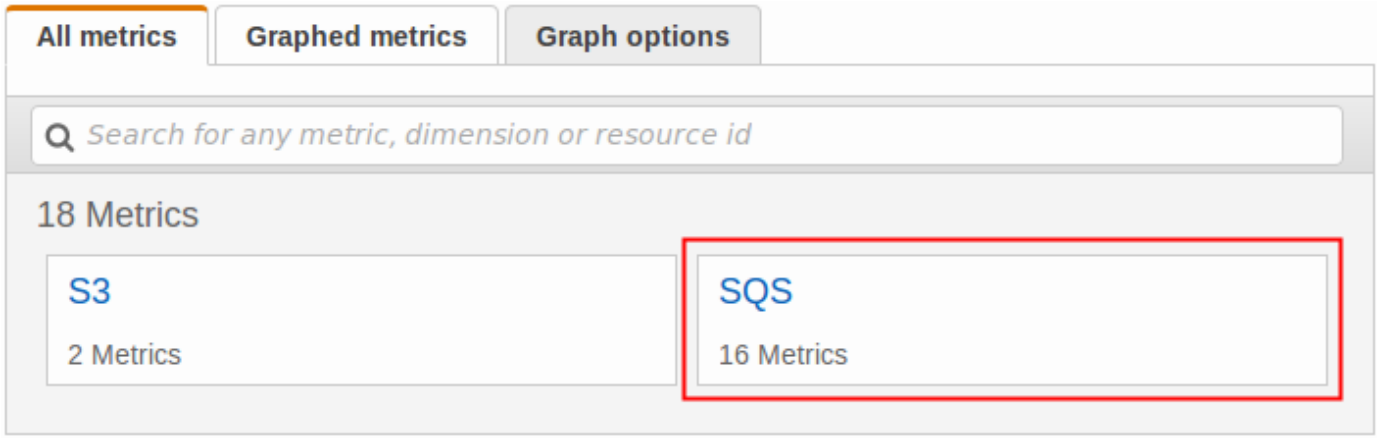
   Vários gráficos são exibidos na seção SQS metrics.
4. Para entender o que um determinado gráfico representa, passe o mouse sobre

   ao lado do gráfico desejado, ou consulte CloudWatch Métricas disponíveis para o Amazon SQS.
5. Para alterar o intervalo de tempo para todos os gráficos ao mesmo tempo, em Time Range, escolha o intervalo de tempo desejado (por exemplo, Last Hour).
6. Para exibir estatísticas adicionais para um gráfico individual, selecione o gráfico.
7. Na caixa de diálogo Detalhes do CloudWatch monitoramento, selecione uma estatística (por exemplo, soma). Para obter uma lista de estáticas suportadas, consulte CloudWatch Métricas disponíveis para o Amazon SQS.

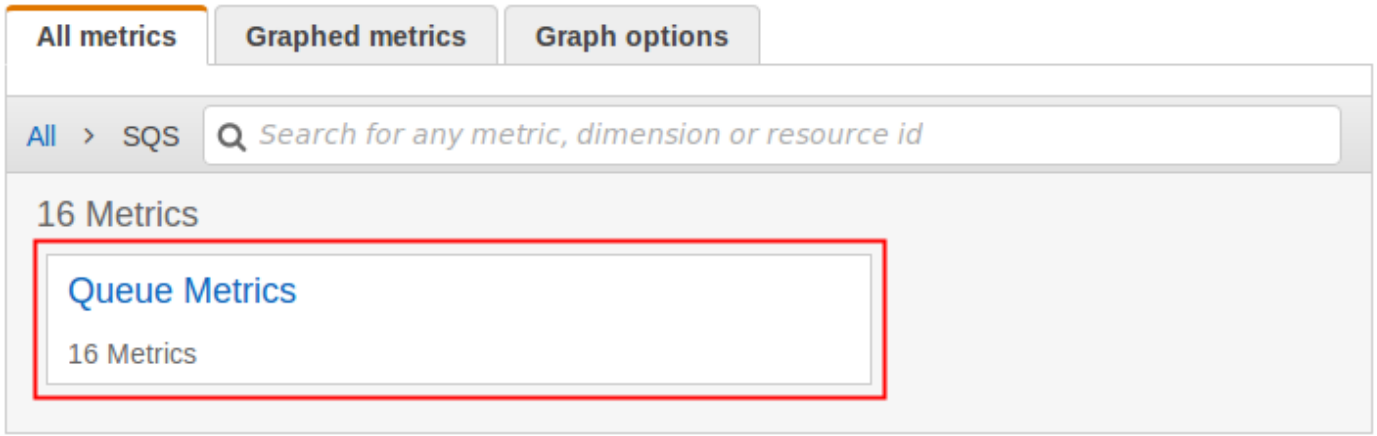
8. Para alterar o intervalo de tempo que um gráfico individual exibe (por exemplo, para mostrar um intervalo de tempo das últimas 24 horas em vez dos últimos 5 minutos, ou mostrar um período de hora em vez de um período de 5 minutos), com a caixa de diálogo do gráfico ainda exibida, em Time Range, escolha o intervalo de tempo desejado (por exemplo, Last 24 Hours). Em Period, escolha o período desejado no intervalo de tempo especificado (por exemplo, 1 Hour). Ao terminar de visualizar o gráfico, clique em Close.
9. (Opcional) Para trabalhar com CloudWatch recursos adicionais, na guia Monitoramento, escolha Exibir todas as CloudWatch métricas e siga as instruções do CloudWatch Console Amazon procedimento.

CloudWatch Console Amazon

1. Faça login no console do CloudWatch.
2. No painel de navegação, selecione Métricas.
3. Selecione o namespace de métrica do SQS.

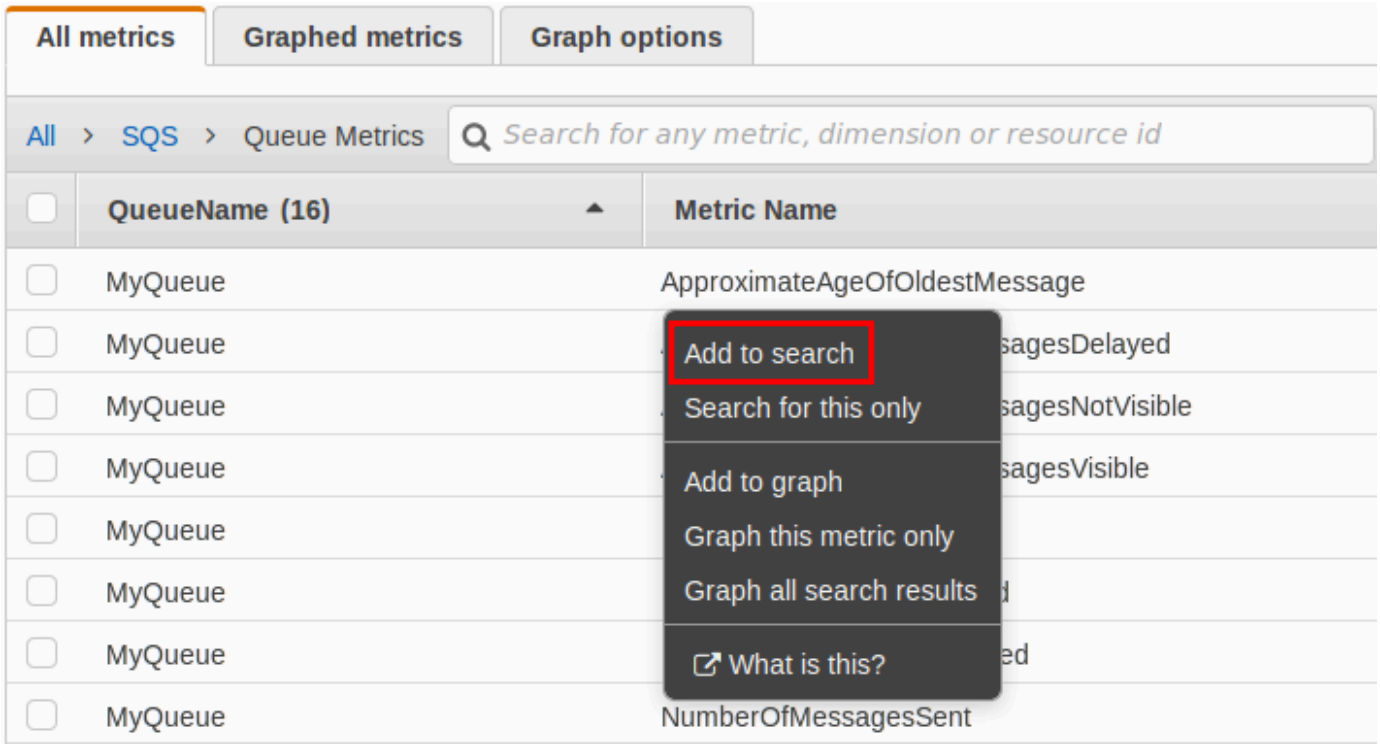


4. Selecione a dimensão de métrica Queue Metrics.



5. Agora você pode examinar as métricas do Amazon SQS:

   - Para classificar a métrica, use o cabeçalho da coluna.
   - Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica.
   - Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Adicionar à pesquisa.



Para obter mais informações e opções adicionais, consulte Métricas gráficas e Uso de CloudWatch painéis da Amazon no Guia do CloudWatch usuário da Amazon.

AWS Command Line Interface

Para acessar as métricas do Amazon SQS usando o AWS CLI, execute o `get-metric-statistics` comando.

Para obter mais informações, consulte Obter estatísticas de uma métrica no Guia CloudWatch do usuário da Amazon.

CloudWatch API

Para acessar as métricas do Amazon SQS usando a CloudWatch API, use a `GetMetricStatistics` ação.

Para obter mais informações, consulte Obter estatísticas de uma métrica no Guia CloudWatch do usuário da Amazon.

Criação de CloudWatch alarmes para métricas do Amazon SQS

CloudWatch permite que você acione alarmes com base em um limite métrico. Por exemplo, você pode criar um alarme para a métrica `NumberOfMessagesSent`. Por exemplo, se mais de 100 mensagens são enviadas à fila `MyQueue` em 1 hora, uma notificação por e-mail é enviada. Para obter mais informações, consulte Criação de CloudWatch alarmes da Amazon no Guia do CloudWatch usuário da Amazon.

1. Faça login no AWS Management Console e abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Alarmes e, em seguida, Criar alarme.
3. Na seção Select Metric (Selecionar métrica) da caixa de diálogo Create Alarm (Criar alerta), selecione Browse Metrics (Procurar métricas), SQS.
4. Para SQS > Queue Metrics, escolha o QueueNamenome da métrica para a qual definir um alarme e, em seguida, escolha Avançar. Para ver uma lista das métricas disponíveis, consulte CloudWatch Métricas disponíveis para o Amazon SQS.

   No exemplo a seguir, a seleção é de um alarme para a métrica `NumberOfMessagesSent` para a fila `MyQueue`. O alarme é acionado quando o número de mensagens enviadas excede 100.
5. Na seção Define Alarm (Definir alerta) da caixa de diálogo Create Alarm (Criar alerta), faça o seguinte:

   a. Em Alarm Threshold (Limite de alerta), digite Name (Nome) e Description (Descrição) para o alerta.

   b. Defina is como > 100.

   c. Defina for (para) como 1 out of 1 datapoints (1 de 1 ponto de dados).

   d. Em Alarm preview (Visualização do alerta), defina Period (Período) como 1 Hour (1 hora).

   e. Defina Statistic (Estatística) como Standard (Padrão), Sum (Soma).

f. Em Actions (Ações), defina Whenever this alarm (Sempre que esse alerta) como State is ALARM (Estado é ALERTA).

   Se você quiser CloudWatch enviar uma notificação quando o alarme for acionado, selecione um tópico existente do Amazon SNS ou escolha Nova lista e insira endereços de e-mail separados por vírgulas.

   > **Note**
   >
   > Se você criar um novo tópico do Amazon SNS, os endereços de e-mail deverão ser verificados para que recebam notificações. Se o estado de alarme for alterado antes que os endereços de e-mail sejam verificados, as notificações não serão recebidas.

6. Escolha Create Alarm.

   O alarme é criado.

CloudWatch Métricas disponíveis para o Amazon SQS

O Amazon SQS envia as seguintes métricas para. CloudWatch

> **Note**
>
> Para filas padrão, o resultado é aproximado por causa da arquitetura distribuída do Amazon SQS. Na maioria dos casos, a contagem deve ser próxima da quantidade real de mensagens na fila.
> Para filas FIFO, o resultado é exato.

Métricas do Amazon SQS

O namespace `AWS/SQS` inclui as métricas a seguir.

Métrica	Descrição
`ApproximateAgeOfOldestMessage`	A idade aproximada de mensagem não excluída mais velha na fila.

Métrica	Descrição
	Note • Depois que uma mensagem é recebida três vezes (ou mais) e não é processada, a mensagem é movida para o final da fila e a métrica `ApproximateAgeOfOldestMessage` aponta para a segunda mensagem mais antiga que não foi recebida mais de três vezes. Essa ação ocorre mesmo que a fila tenha uma política de redirecionamento. • Como uma única mensagem poison-pill (recebida várias vezes, mas nunca excluída) pode distorcer essa métrica, a idade de uma mensagem poison-pill não é incluída na métrica até que a mensagem poison-pill seja consumida com êxito. • Quando a fila tem uma política de redirecionamento, a mensagem é movida para uma dead-letter queue após o número máximo configurado de recebimentos. Quando a mensagem é movida para a dead-letter queue, a métrica

Métrica	Descrição
	`ApproximateAgeOfOldestMessage` da dead-letter queue representa a hora em que a mensagem foi movida para a dead-letter queue (não a hora original em que a mensagem foi enviada). • Para filas FIFO, a mensagem não é movida para o final da fila porque isso quebrará a garantia da ordem da FIFO. Em vez disso, a mensagem será enviada para a DLQ se houver uma configurada. Caso contrário, o grupo de mensagens será bloqueado até que seja excluído com sucesso ou até que expire. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: segundos Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS)

Métrica	Descrição
`ApproximateNumberOfMessagesDelayed`	O número de mensagens na fila que estão atrasadas e indisponíveis para leitura imediata. Isso pode acontecer quando a fila tem a configuração de fila com atraso ou quando uma mensagem foi enviada com um parâmetro de atraso. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: contagem Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS)
`ApproximateNumberOfMessagesNotVisible`	O número de mensagens que estão em trânsito. As mensagens são consideradas como em processamento quando foram enviadas a um cliente, mas ainda não foram excluídas ou ainda não atingiram o final de sua janela de visibilidade. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: contagem Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS)

Métrica	Descrição
`ApproximateNumberOfMessagesVisible`	O número de mensagens a serem processadas. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: contagem Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS) Não há limite no número de mensagens para processar, mas você pode submeter essa lista de pendências a um período de retenção.
`NumberOfEmptyReceives` [1]	O número de chamadas de API `ReceiveMessage` que não retornaram uma mensagem. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: contagem Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS)

Métrica	Descrição
`NumberOfMessagesDeleted` [1]	O número de mensagens excluídas da fila. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: contagem Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS) O Amazon SQS emite a métrica `NumberOfMessagesDeleted` para cada operação de exclusão bem-sucedida usando um identificador de recebimento válido, incluindo exclusões duplicadas. Os seguintes cenários podem fazer com que o valor da métrica `NumberOfMessagesDeleted` seja superior ao esperado: • Acionando a ação `DeleteMessage` nos diversos identificadores de recebimento que pertencem à mesma mensagem: se a mensagem não for processada antes de o tempo limite de visibilidade expirar, a mensagem se torna disponível para outros clientes que podem processá-la excluir novamente, aumentando o valor da métrica `NumberOfMessagesDeleted` . •

Métrica	Descrição
	Acionando a ação `DeleteMessage` no mesmo identificador de recebimento: se a mensagem for processada e excluída, mas você acionar uma ação `DeleteMessage` usando o mesmo identificador de recebimento novamente, um status de sucesso será retornado, aumentando o valor da métrica `NumberOfMessagesDeleted`.
`NumberOfMessagesReceived` [1]	O número de mensagens retornadas por chamadas para a ação de `ReceiveMessage`. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: contagem Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS)

Métrica	Descrição
`NumberOfMessagesSent` [1]	O número de mensagens adicionadas a uma fila. Se você enviar uma mensagem para uma dead letter queue manualmente, ela será capturada pela métrica `NumberOfMessagesSent`. No entanto, se uma mensagem for enviada para uma fila de mensagens mortas como resultado de uma tentativa de processamento malsucedida, ela não será capturada por essa métrica. Assim, é possível que os valores de `NumberOfMessagesSent` e `NumberOfMessagesReceived` sejam diferentes. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidades: contagem Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS)

Métrica	Descrição
`SentMessageSize` [1]	O tamanho das mensagens adicionadas a uma fila. Critérios de relatórios: um valor não negativo é relatado se a fila estiver ativa. Unidade: bytes Estatísticas válidas: média, mínimo, máximo, soma, amostras de dados (exibidas como contagem de amostra no console do Amazon SQS) **Note** `SentMessageSize` não é exibida como uma métrica disponível no CloudWatch console até que pelo menos uma mensagem seja enviada para a fila correspondente.

[1] Essas métricas são calculadas de uma perspectiva de serviço e podem incluir novas tentativas. Não confie nos valores absolutos dessas métricas, ou use-as para estimar o status atual da fila.

Dimensões para métricas do Amazon SQS

A única dimensão para a qual o Amazon SQS envia é. CloudWatch `QueueName` Isso significa que todas as estatísticas disponíveis são filtradas por `QueueName`.

Validação de compatibilidade para o Amazon SQS

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade

Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

> **Note**
>
> Nem todos Serviços da AWS são elegíveis para a HIPAA. Para mais informações, consulte a Referência dos serviços qualificados pela HIPAA.

- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.

- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência no Amazon SQS

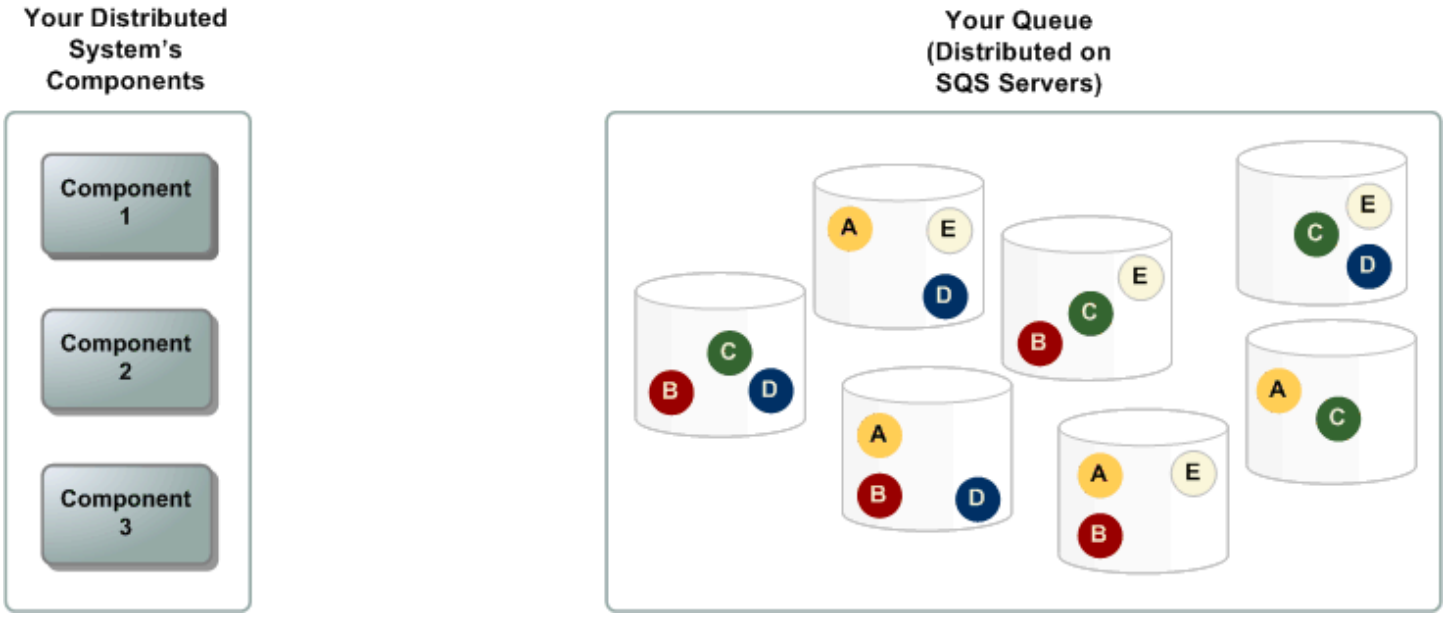
A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas com redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenters tradicionais. Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Além da infraestrutura AWS global, o Amazon SQS oferece filas distribuídas.

Filas distribuídas

Há três partes principais em um sistema de mensagens distribuído: os componentes do sistema distribuído, a fila (distribuída em servidores do Amazon SQS) e as mensagens na fila.

No cenário a seguir, o sistema tem vários produtores (componentes que enviam mensagens para a fila) e consumidores (componentes que recebem mensagens da fila). A fila (que contém as mensagens A a E) armazena as mensagens de forma redundante em vários servidores do Amazon SQS.



Segurança da infraestrutura no Amazon SQS

Como um serviço gerenciado, o Amazon SQS é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa ações de API AWS publicadas para acessar o Amazon SQS pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE).

Você deve assinar as solicitações usando um ID da chave de acesso e uma chave de acesso secreta associados a um principal do IAM. Como alternativa, você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Você pode chamar essas ações de API de qualquer local da rede, mas o Amazon SQS oferece suporte a políticas de acesso com base em recursos, que podem incluir restrições com base no endereço IP de origem. Também é possível usar políticas do Amazon SQS para controlar o acesso de endpoints da Amazon VPC ou de VPCs específicas. Isso isola efetivamente o acesso à rede a uma determinada fila do Amazon SQS somente da VPC específica dentro da rede. AWS Para ter mais informações, consulte Exemplo 5: negar o acesso se não vier de um VPC endpoint.

Práticas recomendadas de segurança para o Amazon SQS

AWS fornece muitos recursos de segurança para o Amazon SQS, que você deve analisar no contexto de sua própria política de segurança. Veja a seguir as práticas recomendadas de segurança preventiva para o Amazon SQS.

> **Note**
>
> As orientações específicas de implementação fornecidas são para casos de uso e implementações comuns. Sugerimos que você examine as melhores práticas no contexto do seu caso de uso, arquitetura e modelo de ameaças específicos.

Tópicos

Garantir que as filas não sejam acessíveis ao público

A menos que você exija explicitamente que qualquer pessoa na Internet possa ler ou gravar na sua fila do Amazon SQS, você deve se certificar de que sua fila não esteja acessível ao público (acessível por qualquer pessoa no mundo ou por qualquer usuário autenticado). AWS

- Evite criar políticas com o `Principal` definido como `""`.
- Evite usar um curinga (`*`). Em vez disso, nomeie um usuário ou usuários específicos.

Implemente o privilégio de acesso mínimo

Quando você concede permissões, você decide quem as recebe, para quais filas as permissões se aplicam, e as ações específicas de API que você deseja permitir para essas filas. A implementação de privilégios mínimos é importante para reduzir os riscos de segurança e o efeito de erros ou intenção maliciosa.

Siga o aviso de segurança padrão de concessão de privilégios mínimos. Ou seja, conceda apenas as permissões necessárias para executar uma tarefa específica. Você pode fazer essa implementação usando uma combinação de políticas de segurança.

O Amazon SQS utiliza o modelo produtor-consumidor, exigindo três tipos de acesso à conta de usuário:

- Administradores: acesso para criar, modificar e excluir filas. Os administradores também controlam as políticas de fila.
- Produtores: acesso para enviar mensagens às filas.
- Consumidores: acesso para receber e excluir mensagens nas filas.

Para obter mais informações, consulte as seções a seguir:

- Gerenciamento de identidade e acesso no Amazon SQS
- Permissões da API do Amazon SQS: referência de ações e recurso
- Usar políticas personalizadas com linguagem de políticas de acesso do Amazon SQS

Use funções do IAM para aplicativos e AWS serviços que exigem acesso ao Amazon SQS

Para que aplicativos ou AWS serviços como o Amazon EC2 acessem as filas do Amazon SQS, eles devem usar credenciais AWS válidas em suas solicitações de API. AWS Como essas credenciais não são alternadas automaticamente, você não deve AWS armazená-las diretamente no aplicativo ou na instância do EC2.

Você deve usar uma função do IAM para gerenciar credenciais temporárias para aplicações ou produtos que precisem acessar o Amazon SQS. Ao usar uma função, você não precisa distribuir credenciais de longo prazo (como nome de usuário, senha e chaves de acesso) para uma instância ou AWS serviço do EC2, como. AWS Lambda Em vez disso, a função fornece permissões temporárias que os aplicativos podem usar quando fazem chamadas para outros AWS recursos.

Para obter mais informações, consulte IAM Roles (Funções do IAM) e Common Scenarios for Roles: Users, Applications, and Services (Cenários comuns para funções: usuários, aplicações e produtos) no Guia do usuário do IAM.

Implemente a criptografia no lado do servidor

Para atenuar problemas de vazamento de dados, utilize a criptografia em repouso para criptografar as mensagens usando uma chave armazenada em um local diferente do local de armazenamento das suas mensagens. A criptografia no lado do servidor (SSE) fornece criptografia dos dados em repouso. O Amazon SQS criptografa os dados no nível da mensagem ao armazená-los e descriptografa as mensagens para você, quando você as acessa. SSE usa chaves gerenciadas em AWS Key Management Service. Se você autenticar sua solicitação e tiver as permissões de acesso, não haverá diferença de acesso entre as filas criptografadas e não criptografadas.

Para obter mais informações, consulte Criptografia em repouso no Amazon SQS e Gerenciamento de chaves do Amazon SQS.

Aplicar a criptografia de dados em trânsito

Sem HTTPS (TLS), um invasor baseado em rede pode espionar o tráfego da rede ou manipulá-lo usando um ataque como o. man-in-the-middle Permita somente conexões criptografadas por HTTPS (TLS), usando a condição `aws:SecureTransport` na política de fila para forçar que as solicitações usem SSL.

Considere usar endpoints da VPC para acessar o Amazon SQS

Se você tiver filas com as quais você deve poder interagir, mas que não devem de forma alguma ficar expostas à Internet, use VPC endpoints para enfileirar o acesso apenas aos hosts dentro de uma VPC específica. Você pode usar políticas de fila para controlar o acesso a filas de endpoints da Amazon VPC específicos ou de VPCs específicas.

Os endpoints da VPC do Amazon SQS fornecem duas maneiras de controlar o acesso às suas mensagens:

- É possível controlar as solicitações, os usuários ou os grupos permitidos por um VPC endpoint específico.
- Você pode controlar quais VPCs ou VPC endpoints terão acesso à fila usando uma política de fila.

Para ter mais informações, consulte Endpoints da Amazon Virtual Private Cloud para o Amazon SQS e Criar uma política de endpoint da Amazon VPC para o Amazon SQS.

Recursos do Amazon SQS relacionados

A tabela a seguir lista os recursos relacionados que serão úteis à medida que você utilizar este serviço.

Recurso	Descrição
Referência da API Amazon Simple Queue Service	Descrições de ações de , parâmetros e tipos de dados, além de uma lista de erros que o serviço retorna.
Amazon SQS na referência de comandos AWS CLI	Descrições dos AWS CLI comandos que você pode usar para trabalhar com filas.
Regiões e endpoints	Informações sobre regiões e endpoints do Amazon SQS
Páginas do produtos	A principal página da Web para obter informações sobre o Amazon SQS.
Fórum de discussão	Um fórum comunitário para que os desenvolvedores discutam questões técnicas relacionadas ao Amazon SQS.
AWS Informações sobre o Premium Support	A principal página da web para obter informações sobre o AWS Premium Support, um canal de suporte individual e de resposta rápida para ajudá-lo a criar e executar aplicativos em serviços de infraestrutura. AWS

Histórico de documentação

A tabela a seguir descreve as alterações importantes feitas no Guia do desenvolvedor do Amazon Simple Queue Service desde janeiro de 2019. Para receber notificações sobre atualizações dessa documentação, inscreva-se no feed RSS.

Às vezes, os recursos do serviço são lançados de forma incremental nas AWS regiões em que um serviço está disponível. Atualizamos esta documentação apenas para a primeira versão. Não fornecemos informações sobre a disponibilidade da região nem anunciamos lançamentos subsequentes da região. Para obter informações sobre a disponibilidade de recursos de serviço na região e para assinar notificações sobre atualizações, consulte O que há de novo em AWS? .

Alteração	Descrição	Data
AWS Protocolo JSON	Faça solicitações de API usando o protocolo AWS JSON.	27 de julho de 2023
Nova seção que descreve as políticas AWS gerenciadas para o Amazon SQS e as atualizações dessas políticas	O Amazon SQS adicionou uma nova ação que permite listar as tarefas mais recentes de movimentação de mensagens (até dez) em uma fila de origem específica. Essa ação está associada à operação de API `ListMessageMoveTasks` .	7 de junho de 2023
Redirecionamento da fila de mensagens não entregues usando APIs	Configure o redirecionamento de filas de mensagens não entregues usando as APIs do Amazon SQS.	7 de junho de 2023
ABAC para o Amazon SQS	Controle de acesso por atributo (ABAC) usando tags de fila para permissões de acesso flexíveis e escaláveis.	10 de novembro de 2022

Aumento do limite de alto throughput de FIFO	Aumento das cotas padrão para o modo de alto throughput de FIFO em regiões comerciais, além da otimização de documentos de alto throughput de FIFO.	20 de outubro de 2022
A criptografia do lado do servidor (SSE) está disponível	Criptografia do lado do servidor (SSE) usando a criptografia do SQS (SSE-SQS) por padrão.	26 de setembro de 2022
A compatibilidade com a proteção confusa para adjunto do Amazon SQS está disponível	A proteção confusa para adjunto permite que você especifique novos cabeçalhos nas respectivas solicitações, que são verificados em relação às condições da política do KMS ao usar a SSE gerenciada pelo Amazon SQS.	29 de dezembro de 2021
A SSE gerenciada está disponível	A SSE gerenciada pelo Amazon SQS (SSE-SQS) é uma criptografia gerenciada do lado do servidor que usa chaves de criptografia de propriedade do SQS para proteger dados sigilosos enviados por filas de mensagens.	23 de novembro de 2021
O redirecionamento da fila de mensagens não entregues está disponível	O Amazon SQS é compatível com o redirecionamento da fila de mensagens não entregues para filas padrão.	10 de novembro de 2021

Disponibilidade de throughput alto para mensagens nas filas FIFO	O alto throughput para filas FIFO do Amazon SQS fornece um maior número de transações por segundo (TPS) para mensagens em filas FIFO. Para obter informaçõ es sobre cotas de taxa de transferência, consulte Cotas relacionadas a mensagens.	27 de maio de 2021
Disponibilidade de throughput alto para mensagens nas filas FIFO na versão de visualiza ção	O alto throughput para filas do Amazon SQS FIFO está na versão de pré-visualização e está sujeita a alterações. Esse recurso proporciona um maior número de transaçõe s por segundo (TPS) para mensagens em filas FIFO. Para obter informações sobre cotas de taxa de transferência, consulte Cotas relacionadas a mensagens.	17 de dezembro de 2020
Novo design do console do Amazon SQS	Para simplificar fluxos de trabalho de desenvolvimento e produção, o console do Amazon SQS apresenta uma nova experiência do usuário.	8 de julho de 2020
O Amazon SQS oferece suporte à paginação para ListQueues e listDeadLetter SourceQueues	Você pode especificar o número máximo de resultado s a serem retornados de uma listQueues ou solicitação de lista. DeadLetter SourceQue ues	22 de junho de 2020

O Amazon SQS oferece suporte a métricas de 1 minuto CloudWatch da Amazon em AWS todas as regiões, exceto AWS GovCloud nas regiões (EUA)	A CloudWatch métrica de um minuto para o Amazon SQS está disponível em todas as regiões, exceto AWS GovCloud (US) nas regiões.	9 de janeiro de 2020
O Amazon SQS oferece suporte a métricas de 1 minuto CloudWatch	Atualmente, a CloudWatch métrica de um minuto para o Amazon SQS está disponíve l somente nas seguintes regiões: Leste dos EUA (Ohio), Europa (Irlanda), Europa (Estocolmo) e Ásia-Pacífico (Tóquio).	25 de novembro de 2019
AWS Lambda gatilhos para filas FIFO do Amazon SQS estão disponíveis	Você pode configurar as mensagens que chegam a uma fila FIFO como acionador es de função Lambda.	25 de novembro de 2019
A criptografia do lado do servidor (SSE) para o Amazon SQS está disponível nas regiões da China	O SSE para Amazon SQS está disponível nas regiões da China.	13 de novembro de 2019
As filas FIFO estão disponíve is na região do Oriente Médio (Bahrein)	As filas FIFO estão disponíve is na região do Oriente Médio (Bahrein).	10 de outubro de 2019
Os endpoints da Amazon Virtual Private Cloud (Amazon VPC) para o Amazon SQS estão disponíveis nas regiões (Leste dos EUA) e AWS GovCloud (Oeste dos EUA) AWS GovCloud	Você pode enviar mensagens para suas filas do Amazon SQS a partir da Amazon VPC nas regiões (Leste dos EUA) e AWS GovCloud (Oeste dos EUA). AWS GovCloud	5 de setembro de 2019

O Amazon SQS permite a solução de problemas de filas AWS X-Ray usando atributos do sistema de mensagens	É possível solucionar problemas de mensagens transmitidas por filas do Amazon SQS usando o X-Ray. Esta versão adiciona o parâmetro de solicitação `MessageSystemAttribute` (que permite enviar cabeçalhos de rastreamento do X-Ray pelo Amazon SQS) às operações de API `SendMessage` e `SendMessageBatch`, o atributo `AWSTraceHeader` à operação de API `ReceiveMessage` e o tipo de dado `MessageSystemAttributeValue`.	28 de agosto de 2019
É possível etiquetar filas do Amazon SQS após a criação	Você pode usar uma única chamada de API do Amazon SQS, função AWS SDK ou comando AWS Command Line Interface (AWS CLI) para criar simultaneamente uma fila e especificar suas tags. Além disso, o Amazon SQS oferece suporte às chaves `aws:TagKeys` e `aws:RequestTag` AWS Identity and Access Management (IAM).	22 de agosto de 2019

O cliente de fila temporária para Amazon SQS já está disponível	Filas temporárias ajudam você a economizar tempo de desenvolvimento e custos de implantação ao usar padrões comuns de mensagens, como solicitação-resposta. Você pode usar o Temporary Queue Client para criar filas temporárias de alta taxa de transferência, econômicas e gerenciadas por aplicações.	25 de julho de 2019
O SSE para Amazon SQS está disponível na região AWS GovCloud (Leste dos EUA)	A criptografia do lado do servidor (SSE) para o Amazon SQS está disponível na região (Leste dos AWS GovCloud EUA).	20 de junho de 2019
As filas FIFO estão disponíveis nas regiões Ásia-Pacífico (Hong Kong), China (Pequim), AWS GovCloud (Leste dos EUA) e AWS GovCloud (Oeste dos EUA)	As filas FIFO estão disponíveis nas regiões Ásia-Pacífico (Hong Kong), China (Pequim), AWS GovCloud (Leste dos EUA) e AWS GovCloud (Oeste dos EUA).	15 de maio de 2019
As políticas de endpoint da Amazon VPC estão disponíveis para o Amazon SQS	Você pode criar políticas de endpoint da Amazon VPC para o Amazon SQS	4 de abril de 2019
As filas FIFO estão disponíveis nas regiões Europa (Estocolmo) e China (Ningxia)	As filas FIFO estão disponíveis nas regiões Europa (Estocolmo) e China (Ningxia).	14 de março de 2019

As filas FIFO estão disponíveis em todas as regiões em que o Amazon SQS está disponível	As filas FIFO estão disponíveis nas regiões Leste dos EUA (Ohio), Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon), Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Singapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Canadá (Central), Europa (Frankfurt), Europa (Irlanda), Europa (Londres), Europa (Paris) e América do Sul (São Paulo).	7 de fevereiro de 2019

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.