
Amazon Simple Queue Service

Guia do desenvolvedor



Amazon Simple Queue Service: Guia do desenvolvedor

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

As marcas comerciais e o visual comercial da Amazon não podem ser usados em conexão com nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa causar confusão entre os clientes ou que deprecie ou desacredite a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, conectados ou patrocinados pela Amazon.

Table of Contents

O que é o Amazon SQS?	1
Benefícios do uso do Amazon SQS	1
Diferenças entre o Amazon SQS, o Amazon MQ e o Amazon SNS	2
Tipos de fila	2
Tarefas comuns para começar a usar o Amazon SQS	3
Definição de preços do Amazon SQS	3
Configuração	4
Etapa 1: Criar uma conta da AWS	4
Etapa 2: Criar um usuário do IAM	4
Etapa 3: Obtenha seu ID de chave de acesso e sua chave de acesso secreta	5
Etapa 4: Prepare-se para usar o código de exemplo	6
Próximas etapas	6
Conceitos básicos	7
Prerequisites	7
Etapa 1: Criar uma fila	7
Etapa 2: Enviar uma mensagem	7
Etapa 3: Receber e excluir sua mensagem	8
Etapa 4: Excluir sua fila	9
Próximas etapas	9
Configurando o Amazon SQS	10
Noções básicas sobre o console do Amazon SQS	10
Antevisão: Alta taxa de transferência para filas FIFO	11
Criação de uma fila	12
Editar uma fila	13
Configurar parâmetros da fila	14
Configurar a política de acesso	15
Configurar o SSE para uma fila	16
Configurar uma fila de mensagens mortas	16
Configurar tags para uma fila	17
Inscrever uma fila em um tópico	18
Configurar um trigger do Lambda	18
Prerequisites	19
Gerenciamento de filas	20
Enviar mensagens	20
Atributos de mensagem	21
Receber e excluir uma mensagem	22
Limpar uma fila	22
Excluir uma fila	23
Confirmando que uma fila está vazia	23
Como o Amazon SQS	25
Arquitetura básica do	25
Filas distribuídas	25
Ciclo de vida de mensagens	26
Filas padrão	28
Ordenação de mensagens	28
Entrega pelo menos uma vez	28
Filas FIFO	28
Ordenação de mensagens	29
Principais termos	30
Lógica da entrega FIFO	30
Processamento exatamente uma vez	31
Mudança de uma fila Padrão para uma fila FIFO	32
Compatibility	32
Identificadores de filas e mensagens	33

Identificadores para filas padrão e FIFO	33
Identificadores adicionais para filas FIFO	34
Metadados de mensagens	35
Atributos de mensagem	35
Atributos do sistema de mensagens	37
Recursos necessários para processar mensagens do	38
Listar filas	38
Tags de alocação de custos	39
Sondagem curta e longa	39
Consumo de mensagens usando sondagem curta	40
Consumo de mensagens usando sondagem longa	40
Diferenças entre as sondagens longa e curta	41
Filas de mensagens mortas	41
Como as filas de mensagens mortas funcionam?	42
Quais são os benefícios das filas de mensagens mortas?	42
Como os diferentes tipos de fila lidam com falhas de mensagens?	43
Quando devo usar uma fila de mensagens mortas?	43
Solução de problemas de filas de mensagens mortas	44
Tempo limite de visibilidade	44
Mensagens em trânsito	45
Definição do tempo limite de visibilidade	46
Alteração do tempo limite de visibilidade de uma mensagem	46
Término do tempo limite de visibilidade de uma mensagem	47
Filas de atraso	47
Filas temporárias	48
Filas virtuais	48
Padrão de mensagens de resposta a solicitação (filas virtuais)	49
Cenário de exemplo: Processamento de uma solicitação de login	50
Limpeza das filas	51
Temporizadores de mensagens	52
Práticas recomendadas	53
Recomendações para filas padrão e FIFO	53
Trabalhar com mensagens do	53
Redução de custos do	55
Mudança de uma fila Padrão para uma fila FIFO	56
Recomendações adicionais para filas FIFO	56
Uso do ID de eliminação de duplicação de mensagens do	57
Uso do ID do grupo de mensagens do	58
Uso do ID de tentativa de solicitação de recebimento do	59
Exemplos do Java SDK	60
Usando criptografia no lado do servidor	60
Adicionando SSE a uma fila existente	60
Desativando o SSE para uma fila	61
Criação de uma fila com SSE	61
Recuperando atributos do SSE	61
Configurar tags	62
Listar tags	62
Adição ou atualização de tags	62
Remover tags	63
Envio de atributos de mensagens	63
Definindo atributos	63
Enviar uma mensagem com atributos	64
Gerenciamento de mensagens grandes	65
Prerequisites	65
Exemplo: Usando o Amazon S3 para gerenciar mensagens grandes do Amazon SQS	65
Como trabalhar com o JMS	69
Prerequisites	69

Introdução à Biblioteca de Mensagens Java	70
Criação de uma conexão JMS	70
Criação de uma fila do Amazon SQS	71
Envio de mensagens de forma síncrona	71
Recebimento de mensagens de forma síncrona	72
Recebimento de mensagens de forma assíncrona	73
Uso do modo de reconhecimento do cliente	74
Uso do modo de reconhecimento não ordenado	75
Usando o cliente JMS com outros clientes do Amazon SQS	76
Java funcional de exemplo para uso do JMS com filas do Amazon SQS Standard	76
ExampleConfiguration.java	77
TextMessageSender.java	78
SyncMessageReceiver.java	80
AsyncMessageReceiver.java	81
SyncMessageReceiverClientAcknowledge.java	82
SyncMessageReceiverUnorderedAcknowledge.java	85
SpringExampleConfiguration.xml	87
SpringExample.java	88
ExampleCommon.java	90
Implementações JMS 1.1 com suporte	91
Interfaces comuns com suporte	91
Tipos de mensagens com suporte	91
Modos de reconhecimento de mensagens com suporte	91
Cabeçalhos definidos pelo JMS e propriedades reservadas	91
Tutoriais	93
Criação de uma fila do Amazon SQS (AWS CloudFormation)	93
Enviar uma mensagem a partir de uma VPC	94
Etapa 1: Criar um key pair do Amazon EC2	95
Etapa 2: Criar recursos da AWS	95
Etapa 3: Confirme que sua instância do EC2 não é acessível publicamente	96
Etapa 4: Criar um Amazon VPC endpoint para o Amazon SQS	97
Etapa 5: Envie uma mensagem para sua fila do Amazon SQS	97
Cotas	99
Cotas relacionadas a filas	99
Cotas relacionadas a mensagens	100
Cotas relacionadas a políticas	102
Automação e solução de problemas	103
Automação de notificações usando o EventBridge	103
Solução de problemas de filas usando X-Ray	103
Segurança	105
Proteção de dados	105
Criptografia de dados	106
Privacidade do tráfego entre redes	112
Identity and Access Management	114
Authentication	114
Controle de acesso	115
Visão geral	116
Usar políticas baseadas em identidade	121
Usar políticas personalizadas com a linguagem da política de acesso	129
Uso de credenciais de segurança temporárias	139
Referência de permissões da API	141
Registro em log e monitoramento	143
Registrar em log as chamadas de API usando o	144
Monitorar filas usando o CloudWatch	148
Validação de conformidade	156
Resiliência	156
Filas distribuídas	156

Segurança da infraestrutura	157
Práticas recomendadas	157
Melhores práticas preventivas	158
Como trabalhar com APIs do	160
Fazer solicitações da API de consulta	160
Criar um endpoint	160
Como fazer uma solicitação GET	161
Como fazer uma solicitação POST	161
Autenticação de solicitações	162
Interpretação de respostas	165
Ações em lote	166
Habilitar o buffer no lado do cliente e o processamento de solicitações em lotes	167
Aumento da taxa de transferência usando escalabilidade horizontal e processamento de ações em lotes	171
Recursos relacionados	181
Histórico de documentação	182
Glossário da AWS	185
.....	clxxxvi

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 construções comuns, como [Filas de mensagens mortas do \(p. 41\)](#) e [Tags de alocação de custos do \(p. 39\)](#). Ele fornece uma API de web services genérica que você pode acessar usando qualquer linguagem de programação compatível com o AWS SDK.

O Amazon SQS [Padrão \(p. 28\)](#) e [Filas FIFO \(p. 28\)](#). Para obter mais informações, consulte [Tipos de fila \(p. 2\)](#).

Tópicos

- [Benefícios do uso do Amazon SQS \(p. 1\)](#)
- [Diferenças entre o Amazon SQS, o Amazon MQ e o Amazon SNS \(p. 2\)](#)
- [Tipos de fila \(p. 2\)](#)
- [Tarefas comuns para começar a usar o Amazon SQS \(p. 3\)](#)
- [Definição de preços do Amazon SQS \(p. 3\)](#)

Benefícios do uso do Amazon SQS

- **Segurança:** [Você controla \(p. 114\)](#) Quem pode enviar e receber mensagens em uma fila do Amazon SQS.
[criptografia do lado do servidor \(SSE\) \(p. 106\)](#) O permite transmitir dados confidenciais protegendo o conteúdo das mensagens nas filas usando chaves gerenciadas no AWS Key Management Service (AWS KMS).
- **Durabilidade-** Para garantir a segurança de suas mensagens, o Amazon SQS as armazena em vários servidores. Suporte a filas padrão [Entrega de mensagem pelo menos uma vez \(p. 28\)](#) e as filas FIFO suportam [processamento de mensagens exatamente uma vez \(p. 31\)](#).
- **Disponibilidade-** O Amazon SQS [Infraestrutura redundante \(p. 25\)](#) Para fornecer acesso altamente simultâneo às mensagens e alta disponibilidade para produzir e consumir mensagens.
- **Escalabilidade—** O Amazon SQS pode processar cada [Solicitação armazenada em buffer \(p. 167\)](#) Independentemente do, 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** do Suas filas não precisam ser exatamente iguais - por exemplo, você pode [definir um atraso padrão em uma fila \(p. 47\)](#). Você pode armazenar o conteúdo de mensagens maiores que 256 KB [Usando o Amazon Simple Storage Service \(Amazon S3\) \(p. 65\)](#) O Amazon DynamoDB, com o Amazon SQS segurando um ponteiro para o objeto Amazon S3, ou você pode dividir uma mensagem grande em mensagens menores.

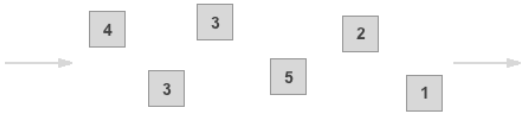

Diferenças entre o Amazon SQS, o Amazon MQ e o Amazon SNS

Amazon SQS e [Amazon SNS](#) são serviços de fila e tópico altamente escaláveis, simples de usar e não exigem a configuração de agentes de mensagens. Recomendamos esses serviços para novos aplicativos que podem se beneficiar de uma escalabilidade praticamente ilimitada e de APIs simples.

[Amazon MQ](#) é um serviço gerenciado de agente de mensagens que fornece compatibilidade com muitos operadores de mensagens populares. Recomendamos o Amazon MQ para migrar aplicativos de operadores de mensagens existentes que dependem da compatibilidade com APIs (como o JMS) ou com protocolos (como AMQP, MQTT, OpenWire e STOMP).

Tipos de fila

A tabela a seguir descreve os recursos das filas padrão e das filas FIFO.

Fila padrão	Fila FIFO
<p>Taxa de transferência ilimitadaAs filas padrão oferecem suporte a um número quase ilimitado de chamadas de API por segundo, por ação de API (<code>SendMessage</code>, <code>ReceiveMessage</code>, ou <code>DeleteMessage</code>).</p> <p>Entrega pelo menos uma vezUma mensagem é entregue pelo menos uma vez, mas às vezes mais de uma cópia da mensagem é entregue.</p> <p>Melhor ordenação do- Às vezes, as mensagens são entregues em uma ordem diferente da qual elas foram enviadas.</p>	<p>Alta taxa de transferência— Se você usar olotes (p. 166)As filas FIFO oferecem suporte a até 3.000 transações por segundo, por método de API (<code>SendMessageBatch</code>, <code>ReceiveMessage</code>, ou <code>DeleteMessageBatch</code>). As 3000 transações 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 (<code>SendMessage</code>, <code>ReceiveMessage</code> ou <code>DeleteMessage</code>).</p> <p>Processamento exatamente uma vezUma mensagem é entregue uma vez e permanece disponível até que um consumidor a processe e exclua. As duplicações não são introduzidas na fila.</p> <p>Entrega primeiro-em-primeiro a sairA ordem em que as mensagens são enviadas e recebidas é preservada estritamente.</p>
	
<p>Enviar dados entre aplicativos quando a taxa de transferência for importante, por exemplo:</p> <ul style="list-style-type: none">Desacoplar solicitações de usuário em tempo real do intenso trabalho em segundo plano: permite que os usuários façam upload de mídia enquanto a redimensionam ou a codificam.	<p>Enviar dados entre aplicativos quando a ordem dos eventos for importante, por exemplo:</p> <ul style="list-style-type: none">Verifique se os comandos inseridos pelo usuário são executados na ordem correta.

Fila padrão	Fila FIFO
<ul style="list-style-type: none">• Alocar tarefas para vários nós operadores: 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.	<ul style="list-style-type: none">• 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.

Tarefas comuns para começar a usar o Amazon SQS

- Para criar sua primeira fila com o Amazon SQS e enviar, receber e excluir uma mensagem, consulte [Conceitos básicos Amazon SQS \(p. 7\)](#).
- Para acionar uma função do Lambda, consulte [Configurar uma fila do para acionar uma função do AWS Lambda \(console\) \(p. 18\)](#).
- Para saber mais sobre a funcionalidade e a arquitetura do Amazon SQS, consulte [Como o Amazon SQS \(p. 25\)](#).
- Para conhecer as diretrizes e advertências que ajudarão você a aproveitar ao máximo o Amazon SQS, consulte [Melhores práticas do Amazon SQS \(p. 53\)](#).
- Explore os exemplos do Amazon SQS para um dos AWS SDKs, como o [Guia do desenvolvedor do AWS SDK for Java 2.x](#).
- Para saber mais sobre as ações do Amazon SQS, consulte o [Referência da API do Amazon Simple Queue Service](#).
- Para saber mais sobre os comandos da CLI da AWS do Amazon SQS, consulte o [Referência de comandos da CLI da AWS](#).

Definição de preços do Amazon SQS

O Amazon SQS não tem custos iniciais. Os primeiros milhões de pedidos mensais são gratuitos. Depois disso, você paga com base no número e no conteúdo das solicitações e nas interações com o Amazon S3 e o AWS Key Management Service.

Para obter mais informações, consulte [Definição de preços do Amazon SQS](#).

Configurar o Amazon SQS

Tópicos

- [Etapa 1: Criar uma conta da AWS](#) (p. 4)
- [Etapa 2: Criar um usuário do IAM](#) (p. 4)
- [Etapa 3: Obtenha seu ID de chave de acesso e sua chave de acesso secreta](#) (p. 5)
- [Etapa 4: Prepare-se para usar o código de exemplo](#) (p. 6)
- [Próximas etapas](#) (p. 6)

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

Etapa 1: Criar uma conta da AWS

Para acessar qualquer serviço da AWS, primeiro é necessário criar um [Conta da AWS](#), uma conta da Amazon.com que pode usar produtos da AWS. Você pode usar sua conta da AWS para visualizar seus relatórios de atividades e de uso e gerenciar autenticação e acesso.

Para evitar o uso do usuário raiz da sua conta da AWS 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.

Para configurar uma nova conta

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções online.

Parte do procedimento de cadastro envolve uma chamada telefônica e a digitação de um código de verificação usando o teclado do telefone.

Etapa 2: Criar um usuário do IAM

Para criar um usuário administrador para você mesmo e adicionar o usuário a um grupo de administradores (console)

1. Faça login no [console do IAM](#) como o proprietário da conta escolhendo Root user (Usuário raiz) e inserindo seu endereço de e-mail da conta da AWS. Na próxima página, insira sua senha.

Note

Recomendamos seguir as melhores práticas para utilizar o **Administrator** Usuário do IAM que segue e armazene as credenciais do usuário raiz com segurança. Cadastre-se como o usuário raiz apenas para executar algumas [tarefas de gerenciamento de serviços e contas](#).

2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Em User name (Nome do usuário), digite **Administrator**.
4. Marque a caixa de seleção ao lado de AWS Management Console access (Acesso ao Console de Gerenciamento da AWS). Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.
5. (Opcional) Por padrão, a AWS exige que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next

sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.

6. Selecione **Próximo: Permissões**
7. Em **Set permissions** (Conceder permissões), escolha **Add user to group** (Adicionar usuário ao grupo).
8. Escolha **Create group** (Criar grupo).
9. Na caixa de diálogo **Create group** (Criar grupo), em **Group name** (Nome do grupo), digite **Administrators**.
10. Selecione **Políticas de filtro** e depois selecione **AWS gerenciado — função de trabalho** para filtrar o conteúdo da tabela.
11. Na lista de políticas, marque a caixa de seleção **AdministratorAccess**. A seguir escolha **Criar grupo**.

Note

Ative acesso do usuário e da função do IAM ao Faturamento para poder usar as permissões de **AdministratorAccess** para acessar o console de Gerenciamento de custos e faturamento da AWS. Para fazer isso, siga as instruções na [etapa 1 do tutorial sobre como delegar acesso ao console de faturamento](#).

12. Suporte a lista de grupos, selecione a caixa de seleção para seu novo grupo. Escolha **Refresh** (Atualizar) caso necessário, para ver o grupo na lista.
13. Selecione **Próximo: Tags**.
14. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre como usar tags no IAM, consulte [Marcar entidades do IAM](#) no Guia do usuário do IAM.
15. Selecione **Próximo: Review** (Revisar) Para ver a lista de associações a grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, selecione **Criar usuário**.

É possível usar esse mesmo processo para criar mais grupos e usuários e conceder aos usuários acesso aos recursos da conta da AWS. Para saber como usar políticas para restringir as permissões de usuário a recursos específicos da AWS, consulte [Gerenciamento de acesso](#) e [Políticas de exemplo](#).

Etapa 3: Obtenha seu ID de chave de acesso e sua chave de acesso secreta

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

Note

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

As chaves de acesso consistem em um ID de chave de acesso e uma chave de acesso secreta, usados para assinar solicitações programáticas feitas por você à AWS. Se você não tiver chaves de acesso, você poderá criá-las no Console de Gerenciamento da AWS. Como melhor prática, não utilize as chaves de acesso do usuário raiz da conta da AWS para nenhuma tarefa para a qual ele não seja necessário. Em vez disso, [criar um novo usuário do IAM do administrador](#) com as chaves de acesso para si mesmo.

A única vez que você pode visualizar ou fazer download da chave de acesso secreta é quando você cria as chaves. Não será possível recuperá-las posteriormente. No entanto, você pode criar novas chaves de acesso a qualquer momento. Você também deve ter permissões para executar as ações do IAM necessárias. Para obter mais informações, consulte [Permissões necessárias para acessar os recursos do IAM](#) no Guia do usuário do IAM.

Para criar chaves de acesso para um usuário do IAM

1. Faça login no Console de Gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Usuários.
3. Escolha o nome do usuário cujas chaves de acesso você deseja criar e escolha Credenciais de segurança.
4. No Chaves de acesso Seção do, escolha Criar chave de acesso.
5. Para visualizar o novo key pair de acesso, selecione Mostrar. Você não terá mais acesso à chave de acesso secreta depois que essa caixa de diálogo for fechada. Suas credenciais terão a seguinte aparência:
 - ID de chave de acesso: AKIAIOSFODNN7EXAMPLE
 - Chave de acesso secreta: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
6. Para baixar o par de chaves, escolha Baixar arquivo .csv. Armazene as chaves em um lugar seguro. Você não terá mais acesso à chave de acesso secreta depois que essa caixa de diálogo for fechada.

Mantenha a confidencialidade das chaves para proteger sua conta da AWS e nunca envie-as por e-mail. Não compartilhe as chaves fora da sua organização, mesmo se uma pesquisa parecer vir da AWS ou da Amazon.com. Alguém que legitimamente represente a Amazon jamais pedirá a você sua chave secreta.
7. Depois de baixar o .csv, escolha Fechar. Quando você cria uma chave de acesso, o par de chaves é ativo por padrão, e você pode usar o par imediatamente.

Tópicos relacionados

- [O que é o IAM?](#) no Guia do usuário do IAM
- [Criar credenciais de acesso](#) em Referência geral da AWS

Etapa 4: Prepare-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 aplicativos da AWS em outras linguagens de programação, como Go, JavaScript, Python e Ruby. Para obter mais informações, consulte [Ferramentas para desenvolver e gerenciar aplicativos na AWS](#).

Note

Você pode explorar o Amazon SQS sem escrever código com ferramentas como a Interface da linha de comando da AWS (CLI da AWS) ou o Windows PowerShell. Você pode encontrar exemplos de CLI da AWS no [Seção do Amazon SQS](#) do Referência de comandos da CLI da AWS. Veja exemplos do Windows PowerShell na seção do Amazon Simple Queue Service da [Referência do cmdlet da AWS Tools for PowerShell](#).

Próximas etapas

Você está pronto para [Conceitos básicos \(p. 7\)](#) com o gerenciamento de filas e mensagens do Amazon SQS usando o AWS Management Console.

Conceitos básicos Amazon SQS

Esta seção o ajudará a se tornar mais familiarizado com o Amazon SQS, mostrando como gerenciar filas e mensagens usando o console do Amazon SQS

Tópicos

- [Prerequisites \(p. 7\)](#)
- [Etapa 1: Criar uma fila \(p. 7\)](#)
- [Etapa 2: Enviar uma mensagem \(p. 7\)](#)
- [Etapa 3: Receber e excluir sua mensagem \(p. 8\)](#)
- [Etapa 4: Excluir sua fila \(p. 9\)](#)
- [Próximas etapas \(p. 9\)](#)

Prerequisites

Antes de começar, conclua as tarefas em [Configurar o Amazon SQS \(p. 4\)](#).

Etapa 1: Criar uma fila

A tarefa do Amazon SQS A é a criação de filas. Este procedimento mostra como criar e configurar uma fila FIFO.

1. Abra o console do Amazon SQS <https://console.aws.amazon.com/sqs/>.
2. Escolha Create queue.
3. No Criar fila, especifique a região correta.
4. O Padrão O tipo de fila é selecionado por padrão. Selecione FIFO.

Não possível alterar o tipo de fila após a criação de uma fila.

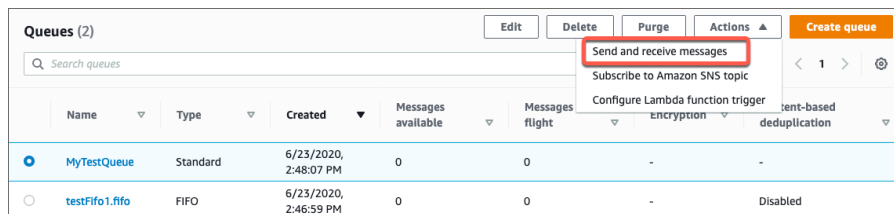
5. Insira um Nome Para sua fila. O nome de uma fila FIFO deve terminar com o `.fifo` Sufixo.
6. Para criar sua fila com os parâmetros padrão, role até a parte inferior e escolha Criar fila. O Amazon SQS cria a fila e exibe a Detalhes.

O Amazon SQS propaga informações sobre a nova fila no sistema. Como o Amazon SQS é um sistema distribuído, você pode enfrentar um pequeno atraso antes que a fila seja exibida no Filas.

Etapa 2: Enviar uma mensagem

Após a criação da fila, você pode enviar uma mensagem para ela.

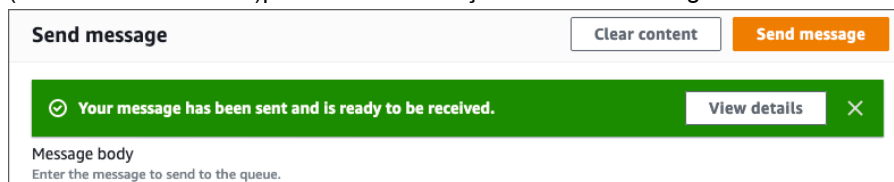
1. No painel de navegação esquerdo, escolha Filas. Na lista de filas, selecione a fila que você criou.
2. No Ações, escolha Enviar e receber mensagens.



O console exibirá oEnviar e receber mensagens.

3. Insira o texto na caixaCorpo da mensagem
4. Insira umID do grupo de mensagensPara a fila. Para obter mais informações, consulte [Lógica da entrega FIFO \(p. 30\)](#).
5. (Opcional) Insira umID de eliminação de duplicação de mensagem. Se você habilitar a deduplicação baseada em conteúdo, o ID de deduplicação da mensagem não será necessário. Para obter mais informações, consulte [Lógica da entrega FIFO \(p. 30\)](#).
6. Escolha Send Message (Enviar mensagem).

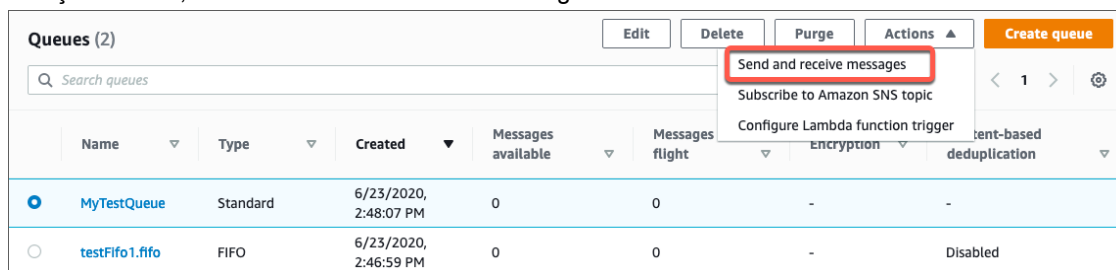
Sua mensagem é enviada e o console exibirá uma mensagem de sucesso. SelecioneView details (Visualizar os detalhes)para exibir informações sobre a mensagem enviada.



Etapa 3: Receber e excluir sua mensagem

Depois de enviar uma mensagem para uma fila, você pode recuperar a mensagem da fila. Ao solicitar mensagens de uma fila, você não pode especificar qual mensagem recuperar. Em vez disso, você especifica o número de mensagens máximo (até 10) que deseja recuperar.

1. DoFilas, selecione uma fila.
2. NoAções da fila, selecioneEnviar e receber mensagens.



O console exibirá oEnviar e receber mensagens.

3. SelecioneEnquete para mensagens.

O Amazon SQS começa a pesquisar servidores para encontrar mensagens na fila. A barra de progresso no lado direito doReceber mensagensexibe a duração da sondagem.

OMensagensexibe 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.

4. Para excluir mensagens, selecione as mensagens que deseja excluir e escolhaExcluir.

5. NoExcluir mensagens, escolhaExcluir.

Etapa 4: Excluir sua fila

1. Na lista de filas, selecione a fila que você criou.
2. DoFilas, selecione a fila a ser excluída.
3. SelecioneExcluir fila.
O console exibirá oExcluir fila.
4. NoExcluir fila, confirme a exclusão inserindo **delete**.
5. Escolha Delete (Excluir).

Próximas etapas

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:

- [Configurar filas, incluindo SSE e outros recursos \(p. 10\)](#).
- [Enviar uma mensagem com atributos. \(p. 21\)](#)
- [Enviar uma mensagem a partir de uma VPC. \(p. 94\)](#)
- Saiba mais sobre os fluxos de trabalho e os processos do Amazon SQS: Leia [Como as filas funcionam \(p. 25\)](#), [Melhores práticas \(p. 53\)](#), e [Cotas \(p. 99\)](#). Você também pode explorar o [Artigos e tutoriais do Amazon SQS](#). Se você tiver alguma dúvida, procure [Perguntas Amazon SQS](#) ou participe do [Fóruns de desenvolvedores do Amazon SQS](#).
- Saiba como interagir com o Amazon SQS. Leia [Trabalhar com APIs \(p. 160\)](#) e explore o [Bibliotecas e código de exemplo](#) e os centros de desenvolvimento:
 - [Java](#)
 - [JavaScript](#)
 - [PHP](#)
 - [Python](#)
 - [Ruby](#)
 - [Windows e .NET](#)
- Saiba mais sobre como acompanhar custos e recursos na seção [Automação e solução de problemas de filas do Amazon SQS \(p. 103\)](#).
- Saiba mais sobre como proteger seus dados e o acesso aos dados na seção [Segurança \(p. 105\)](#).

Configurando filas do Amazon SQS (console)

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 no lado do servidor, associar uma fila de letras mortas à sua fila ou definir um gatilho para invocar uma função do AWS Lambda.

Tópicos

- [Noções básicas sobre o console do Amazon SQS \(p. 10\)](#)
- [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#)
- [Criação de uma fila do Amazon SQS \(console\) \(p. 12\)](#)
- [Editar uma fila do Amazon SQS \(console\) \(p. 13\)](#)
- [Configurando parâmetros de fila \(console\) \(p. 14\)](#)
- [Configurar a política de acesso \(console\) \(p. 15\)](#)
- [Configurar a criptografia no lado do servidor \(SSE\) para uma fila \(console\) \(p. 16\)](#)
- [Configurar uma dead letter queue \(console\) \(p. 16\)](#)
- [Configurando tags de alocação de custo para uma fila do Amazon SQS \(console\) \(p. 17\)](#)
- [Assinar uma fila do Amazon SQS em um tópico do Amazon SNS \(console\) \(p. 18\)](#)
- [Configurar uma fila do para acionar uma função do AWS Lambda \(console\) \(p. 18\)](#)

Noções básicas sobre o console do Amazon SQS

Quando você abrir o console do, selecione Filas no painel de navegação para exibir o Filas. O Filas fornece informações sobre todas as filas na região ativa.

Queues (2)

Edit

Delete

Purge

Actions ▾

Create queue

Q Search queues

< 1 > ⓘ

	Name ▾	Type ▾	Created ▾	Messages available ▾	Messages in flight ▾	Encryption ▾	Content-based deduplication ▾
<input checked="" type="radio"/>	MyTestQueue	Standard	6/23/2020, 2:48:07 PM	0	0	-	-
<input type="radio"/>	testFifo1.fifo	FIFO	6/23/2020, 2:46:59 PM	0	0	-	Disabled

A entrada de cada fila mostra o tipo de fila e outras informações sobre a fila. O Tipo ajuda a distinguir filas padrão das filas FIFO (primeiro a entrar, primeiro a sair) do.

Do 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 na fila.

Queues (2)

Edit

Delete

Purge

Actions ▾

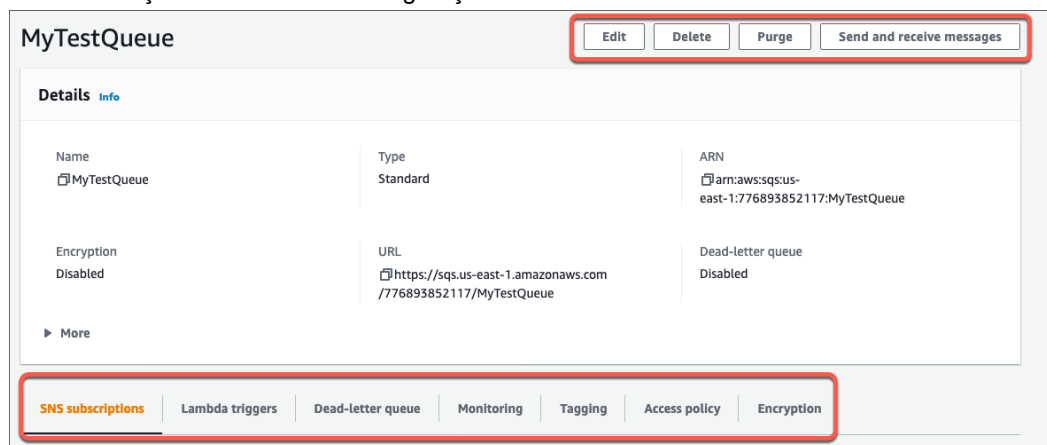
Create queue

Q Search queues

< 1 > ⓘ

	Name ▾	Type ▾	Created ▾	Messages available ▾	Messages in flight ▾	Encryption ▾	Content-based deduplication ▾
<div><input checked="" type="radio"/></div>	MyTestQueue	Standard	6/23/2020, 2:48:07 PM	0	0	-	-
<div><input type="radio"/></div>	testFifo1.fifo	FIFO	6/23/2020, 2:46:59 PM	0	0	-	Disabled

Você também pode escolher o nome da fila, que abre o **Detalhes** para a fila. O **Detalhes** inclui as mesmas ações que a página **Filas**. Além disso, você pode escolher uma das guias abaixo do **Detalhes** para exibir detalhes e ações adicionais de configuração.



Antevisão: Alta taxa de transferência para filas FIFO

Este recurso é oferecido como uma pré-visualização por um tempo limitado (“Pré-visualização”). A AWS pode alterar a funcionalidade da visualização a qualquer momento. A Pré-visualização não se destina a cargas de trabalho de produção e aplicam-se taxas padrão. Seu uso da Visualização é regido pelo [Termos de Serviço da AWS](#), incluindo os termos em Betas e Préviews.

Alta taxa de transferência para [Filas FIFO \(p. 28\)](#) O fornece um número alto de transações por segundo (TPS) para mensagens nessas filas. Para obter informações sobre cotas de taxa de transferência, consulte [Cotas relacionadas a mensagens \(p. 100\)](#).

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

- Habilitar High taxa de transferência FIFO— Torna maior taxa de transferência disponível para mensagens na fila FIFO atual.
- Escopo de deduplicação— Especifica se a deduplicação ocorre no nível da fila ou do grupo de mensagens.
- Limite de transferência FIFO— Especifica se a cota de throughput em mensagens na fila FIFO está definida no nível da fila ou do grupo de mensagens.

No momento, esse recurso está disponível nas seguintes regiões da AWS:

Nome da região	Região
Leste dos EUA (Ohio)	us-east-2
Leste dos EUA (Norte da Virgínia)	us-east-1
Oeste dos EUA (Oregon)	us-west-2

Nome da região	Região
Europa (Irlanda)	eu-west-1

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

1. A partir de uma região suportada, inicie [creating \(p. 12\)](#) ou [Edição \(p. 13\)](#) uma fila FIFO.
2. Ao especificar opções para a fila, escolha **Habilitar High taxa de transferência FIFO**.

Habilitar o FIFO de alta taxa de transferência define as opções relacionadas da seguinte maneira:

- Escopo de deduplicação é definido como **Grupo de mensagens**, a configuração necessária para usar FIFO de alta taxa de transferência.
- Limite de transferência FIFO é definido como **Por ID do grupo de mensagens**, a configuração necessária para usar FIFO de alta taxa de transferência.

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

3. Continue especificando todas as opções para a fila. Ao concluir, selecione **Criação de filas** ou **Save (Salvar)**.

Depois de criar ou editar a fila do FIFO, você pode [enviar mensagens \(p. 20\)](#) a ele e [receber e excluir mensagens \(p. 22\)](#), tudo em um TPS mais alto.

Criação de uma fila do Amazon SQS (console)

Alta taxa de transferência das filas FIFO do Amazon SQS está em versão de teste e sujeito a alterações. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#).

Você pode usar o console do Amazon SQS para criar [filas padrão \(p. 28\)](#) e [Filas FIFO \(p. 28\)](#). O console fornece valores padrão para todas as configurações, exceto para o nome da fila.

Para criar uma fila do Amazon SQS (console)

1. Abra o console do Amazon SQS em <https://console.aws.amazon.com/sqs/>.
2. Escolha **Create queue**.
3. Para o **Tipo**, o **Padrão** tipo de fila é definido por padrão. Para criar uma fila FIFO, escolha **FIFO**.

Note

Você não pode alterar o tipo de fila depois de criar a fila.

4. Digite um **Nome** para a fila. O nome de uma fila FIFO deve terminar com `.fifo` Sufixo.
5. (Opcional) O console define valores padrão para a fila [Parâmetros de configuração \(p. 14\)](#). Under **Configuração**, você pode definir novos valores para os seguintes parâmetros:
 - a. para o **Tempo limite de visibilidade**, insira a duração e as unidades. O intervalo é de 0 a 12 horas. O valor de padrão é de 30 segundos.
 - b. para o **Período de retenção de mensagens**, insira a duração e as unidades. O intervalo é de 1 minuto a 14 dias. O valor padrão é de 4 dias.

- c. para oRetardo de entrega, insira a duração e as unidades. O intervalo é de 0 a 15 minutos. O valor padrão é 0 segundos.
 - d. para oTamanho máximo da mensagem, insira um valor. O intervalo é de 1 KB a 256 KB. O valor padrão é 256 KB.
 - e. para oTempo de espera da mensagem de recebimento, insira um valor. O intervalo é de 0 a 20 segundos. O valor padrão é 0 segundos, o que define [Sondagem curta \(p. 39\)](#). Qualquer valor diferente de zero define sondagem longa.
 - f. Para uma fila FIFO, escolhaAtivar a eliminação de duplicação baseada em conteúdoPara ativar a eliminação de duplicação baseada em conteúdo. A configuração padrão é disabled.
 - g. (Opcional) Para uma fila FIFO, para habilitar uma taxa de transferência mais alta para enviar e receber mensagens na fila, escolhaAtivar FIFO de taxa de transferência alta. Essa opção está atualmente em versão de teste. Escolher esta opção altera as opções de visualização relacionadas (Escopo de eliminação de duplicaçãoeLimite de transferência FIFO) para as configurações necessárias para habilitar a alta taxa de transferência. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#) e [Cotas relacionadas a mensagens \(p. 100\)](#).
6. (Opcional) Defina umaPolítica de acesso. O[política de acesso padrão \(p. 137\)](#) define as contas, usuários e funções que podem acessar a fila. A política de acesso também define as ações (comoSendMessage,ReceiveMessage, ouDeleteMessage) 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, execute uma das seguintes ações:

- SelecioneBasicPara configurar quem pode enviar mensagens para a fila e quem pode receber mensagens da fila. O console cria a política com base em suas escolhas e exibe a política de acesso resultante no painel JSON somente leitura.
 - SelecioneAdvanced (Avançado)para modificar a política de acesso JSON diretamente. Isso permite que você especifique um conjunto personalizado de ações que cada principal (conta, usuário ou função) pode executar.
- 7. (Opcional) Para configurar[Criptografia de \(p. 16\)](#)para a fila, expandaCriptografia.
 - 8. (Opcional) Para configurar um[Fila de mensagens mortas \(p. 16\)](#)para receber mensagens que não podem ser entregues, expandaFila de mensagens mortas.
 - 9. (Opcional) Para adicionar[Tags do \(p. 17\)](#)para a fila, expandaTags.
 - 10. Escolha Create queue. O Amazon SQS cria a fila e exibe aDetalhes.

O Amazon SQS propaga informações sobre a nova fila no sistema. Como o Amazon SQS é um sistema distribuído, você pode enfrentar um pequeno atraso antes que o console exiba a fila noFilas.

Depois de criar uma fila, você pode[enviar mensagens \(p. 20\)](#)a ele, [receber e excluir mensagens \(p. 22\)](#). Você também pode[editar \(p. 13\)](#)qualquer uma das definições de configuração de fila, exceto o tipo de fila.

Editar uma fila do Amazon SQS (console)

O alto throughput das filas FIFO do Amazon SQS está na versão de demonstração e sujeito a alterações. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#).

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 as [Página Filas](#) do console do Amazon SQS.
2. Selecione uma fila e, em seguida, escolha [Edite](#).
3. (Opcional) Em [Configuração](#), atualize o [Parâmetros de configuração \(p. 14\)](#).
 - (Opcional) Para uma fila FIFO, para habilitar uma taxa de transferência mais alta para enviar e receber mensagens na fila, escolha [Habilitar FIFO de alto débito](#). Esta opção está atualmente em versão de demonstração. Escolher esta opção altera as opções de visualização relacionadas (Escopo de deduplicação e Limite de throughput FIFO) para as configurações necessárias para habilitar a alta taxa de transferência. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#) e [Cotas relacionadas a mensagens \(p. 100\)](#).
4. (Opcional) Para atualizar o [política de acesso padrão \(p. 15\)](#), em [Política de acesso](#), modifique a política JSON.
5. (Opcional) Para adicionar, atualizar ou remover [Criptografia de \(p. 16\)](#), expanda [Criptografia](#).
6. (Opcional) Para adicionar, atualizar ou remover um [Fila de mensagens mortas \(p. 16\)](#) (que permite que você receba mensagens não entregues), expanda [Fila de mensagens mortas](#).
7. (Opcional) Para adicionar, atualizar ou remover o [Tags do \(p. 17\)](#) para a fila, expanda [Tags](#).
8. Escolha [Save \(Salvar\)](#).

O console exibe o [Detalhes](#) para a fila.

Configurando parâmetros de fila (console)

Alta taxa de transferência para filas FIFO do Amazon SQS está em versão de teste e sujeito a alterações. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#).

Quando você [create \(p. 12\)](#) ou [Edite \(p. 13\)](#) Uma fila, você poderá configurar os seguintes parâmetros:

- **Tempo limite de visibilidade**— O período de tempo em que uma mensagem recebida de uma fila (por um consumidor) não será visível para os outros consumidores de mensagens. Para obter mais informações, consulte [Tempo limite de visibilidade \(p. 44\)](#).

Note

Usar o console para configurar o tempo limite de visibilidade configura o valor de tempo limite para todas as mensagens na fila. Para configurar o timeout para mensagens únicas ou múltiplas, você deve usar um dos SDKs da AWS.

- **Período de retenção da mensagem**— A quantidade de tempo que o Amazon SQS retém as mensagens que permanecem na fila. Por padrão, a 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 da mensagem](#).
- **Restart**— Quanto tempo o Amazon SQS atrasará antes de enviar uma mensagem adicionada à fila. Para obter mais informações, consulte [Restart \(p. 47\)](#).
- **Tamanho máximo da mensagem**— O tamanho máximo da mensagem para essa fila. Para obter mais informações, consulte [Tamanho máximo da mensagem \(p. 65\)](#).
- **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 obter mais informações, consulte [Sondagem curta e longa do Amazon SQS \(p. 39\)](#).

- Ativar a eliminação de duplicação baseada em conteúdo— O Amazon SQS pode criar automaticamente IDs de deduplicação com base no corpo da mensagem. Para obter mais informações, consulte [Filas FIFO do Amazon SQS \(p. 28\)](#).
- Ativar FIFO de taxa de transferência alta— Use para ativar taxa de transferência alta para mensagens na fila. Essa opção está atualmente em versão de demonstração. Escolher esta opção altera as opções de visualização relacionadas (Escopo de eliminação de duplicação e Limite de transferência FIFO) para as configurações necessárias para habilitar a alta taxa de transferência. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#) e [Cotas relacionadas a mensagens \(p. 100\)](#).

Para configurar parâmetros de fila 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. Escolha uma fila e escolha Edite.
3. Desloque até a Configuração de seção.
4. para o Tempo limite de visibilidade, insira a duração e as unidades. O intervalo é de 0 segundos a 12 horas. O valor de padrão é de 30 segundos.
5. para o Período de retenção da mensagem, insira a duração e as unidades. O intervalo é de 1 minuto a 14 dias. O valor padrão é de 4 dias.
6. para o Restart, insira a duração e as unidades. O intervalo é de 0 segundos a 15 minutos. O valor padrão é 0 segundos.
7. para o Tamanho máximo da mensagem, insira um valor. O intervalo é de 1 KB a 256 KB. O valor padrão é 256 KB.
8. Para uma fila padrão, insira um valor para Tempo de espera da mensagem. O intervalo é de 0 a 20 segundos. O valor padrão é 0 segundos, que define [sondagem curta \(p. 39\)](#). Qualquer valor diferente de zero define sondagem longa.
9. Para obter uma fila FIFO, escolha Ativar a eliminação de duplicação baseada em conteúdo Para habilitar a eliminação de duplicação baseada em conteúdo. A configuração padrão é desativada.
10. (Opcional) Para uma fila FIFO, para habilitar alta taxa de transferência para enviar e receber mensagens na fila, escolha Ativar FIFO de taxa de transferência alta (versão de pré-visualização).
11. Quando terminar de configurar os parâmetros da fila, escolha Save (Salvar).

Configurar a política de acesso (console)

Quando você [Edite \(p. 13\)](#) uma fila, você pode configurar sua política de acesso.

A política de acesso define as contas, os usuários e as 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 Edite.
4. Scroll to the Política de acesso de seção.
5. Edite as instruções de política de acesso na caixa de entrada.

6. Quando terminar de configurar a política de acesso, escolha **Save** (Salvar).

Configurar a criptografia no lado do servidor (SSE) para uma fila (console)

Para proteger os dados nas mensagens da fila, você pode habilitar a criptografia no lado do servidor (SSE) para uma fila. O Amazon SQS integra-se ao AWS Key Management Service (AWS KMS) para gerenciar [chaves mestras do cliente](#) (CMKs) para criptografia no lado do servidor (SSE). Para obter informações sobre como usar a SSE, consulte [Criptografia em repouso](#) (p. 106).

O CMK que você atribui à fila deve ter uma diretiva de chave que inclua permissões para todos os principais autorizados a usar a fila. Para obter mais informações, consulte [Gerenciamento de chaves](#) (p. 108).

Se você não for o proprietário da CMK ou se fizer login com uma conta que não tenha `kms:ListAliases` e `kms:DescribeKey`, você não poderá visualizar as informações sobre a CMK no console do Amazon SQS. Peça ao proprietário da CMK para conceder essas permissões a você. Para obter mais informações, consulte [Gerenciamento de chaves](#) (p. 108).

Quando você [create](#) (p. 12) ou [Edite](#) (p. 13) uma fila, você pode configurar o SSE.

Para configurar o SSE 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 **Edite**.
4. Desloque até **Criptografia**.
5. No **Criptografia**, selecione **Enabled** (Habilitado) para habilitar o SSE.

O console exibe **Descrição**, **Conta**, e **ARN da CMK**.

6. Especifique o ID CMK para a fila. Para obter mais informações, consulte [Principais termos](#) (p. 107).
 - a. Selecione **Escolha um alias CMK**.
 - b. A chave padrão é a CMK gerenciada pela AWS para o Amazon SQS. Para usar essa chave, escolha-a na lista **Key mestra do cliente**.
 - c. Para usar uma CMK personalizada de sua conta da AWS, escolha-a no **Key mestra do cliente**. Para obter instruções sobre como criar CMKs personalizadas, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.
 - d. Para usar um CMK personalizado que não esteja na lista ou um CMK personalizado de outra conta da AWS, escolha **Insira o alias CMK** e insira o nome de recurso da Amazon da CMK (ARN).
7. (Optional) Para **Período de reutilização de chaves de dados**, especifique um valor entre 1 minuto e 24 horas. O padrão é 5 minutos. Para obter mais informações, consulte [Entender o período de reutilização de chaves de dados](#) (p. 111).
8. Ao concluir a configuração do SSE, selecione **Save** (Salvar).

Configurar uma dead letter queue (console)

A **Fila de letras mortas** é uma fila que uma ou mais filas de origem podem usar para mensagens que não são consumidas com êxito. Para obter mais informações, consulte [Filas de mensagens mortas do Amazon SQS](#) (p. 41).

O Amazon SQS não cria a dead letter queue automaticamente. É necessário primeiro criar a fila antes de usá-la como fila de mensagens mortas.

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 [create \(p. 12\)](#) ou [Edite \(p. 13\)](#) 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. Escolha uma fila e escolha [Edite](#).
4. Rolar até o [Fila de letras mortas](#) e escolha [Enabled](#).
5. Escolha o Nome de recurso da Amazon (ARN) de uma [Fila de letras 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 dead letter queue, defina [Maximum receive](#) como um valor entre 1 e 1.000.
7. Quando terminar de configurar a fila de mensagens mortas, escolha [Save](#) (Salvar).

Depois de salvar a fila, o console exibe o [Detalhes](#) para sua fila. No [Detalhes](#), o [Fila de letras mortas](#) exibe a guia [Máximo de Fila de letras mortas ARN na Fila de letras mortas](#).

Configurando tags de alocação de custo para uma fila do Amazon SQS (console)

Para ajudar a organizar e identificar suas filas do Amazon SQS, você pode adicionar tags de alocação de custos a elas. Para obter mais informações, consulte [Tag de alocação de custos do Amazon SQS \(p. 39\)](#).

No [Detalhes](#) para uma fila, o [Atribuição de tags \(tagging\)](#) exibe as marcas para a fila.

Quando você [create \(p. 12\)](#) ou [Edite \(p. 13\)](#) uma fila, você 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 [Edite](#).
4. Desloque-se até [Tags](#) seção.
5. Adicione, modifique ou remova tags de fila:
 - a. Para adicionar uma tag, escolha [Adicionar tag](#) insira um [Chave](#) e [Valor](#), depois, escolha [Adicionar tag](#).
 - b. Para atualizar uma tag, altere sua [Chave](#) e [Valor](#).
 - c. Para remover uma tag, escolha [Remover](#) Ao lado do par de chave-valor.
6. Ao terminar de configurar as tags, escolha [Save](#) (Salvar).

Assinar uma fila do Amazon SQS em um tópico do Amazon SNS (console)

Você pode inscrever 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 uma das filas inscritas. O Amazon SQS gerencia a a a a a e todas as permissões necessárias. Para obter mais informações sobre o Amazon SNS, consulte [O que é o Amazon Simple Notification Service?](#) no 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 do KMS para serviços da AWS](#) (p. 109).

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, selecione a fila que assina o tópico do SNS.
4. No Ações, escolha Assinar o tópico do Amazon SNS.
5. Do Especifique um tópico do Amazon SNS disponível para esta fila. Selecione o tópico do SNS para sua fila.

Se o tópico SNS não estiver listado no menu, escolha Insira o ARN do tópico do Amazon SNS, em seguida, insira o nome de recurso da Amazon (ARN) do tópico.

6. Escolha Save (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 obter mais informações, consulte [Publicar uma mensagem em um tópico do Amazon SNS](#) no Amazon Simple Notification Service.

Se a fila do Amazon SQS e o tópico do SNS estiverem em contas diferentes da AWS, o proprietário do tópico deverá primeiro confirmar a assinatura. Para obter mais informações, consulte [Confirme a assinatura](#) no Amazon Simple Notification Service.

Para obter informações sobre como assinar um tópico de SNS entre regiões, consulte [Envio de mensagens do Amazon SNS para uma fila do Amazon SQS ou função do AWS Lambda em uma região diferente](#) no Amazon Simple Notification Service.

Configurar uma fila do para acionar uma função do AWS Lambda (console)

Você pode usar uma função do AWS Lambda para processar mensagens em uma fila do Amazon SQS. O Lambda sonda a fila e invoca sua função do Lambda sincronicamente com um evento que contém a fila de mensagens. Você pode especificar outra fila para atuar como um Fila de mensagens mortas mortas para mensagens que sua função do Lambda não pode processar.

Uma função do Lambda pode processar itens de várias filas (usando uma origem de eventos do Lambda para cada fila). Você pode usar a mesma fila com várias funções do Lambda.

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

Observe as seguintes restrições:

- Sua fila e a função do Lambda devem ser na mesma região da AWS.
- Uma [fila criptografada \(p. 106\)](#) que usa a chave padrão (CMK gerenciado pela AWS para Amazon SQS) não pode invocar uma função do Lambda em uma conta da AWS diferente.

Para obter informações sobre como implementar a função do Lambda, consulte [Usar o AWS Lambda com o Amazon SQS](#) no Guia do desenvolvedor AWS Lambda.

Prerequisites

Para configurar acionadores de função do Lambda, é necessário atender aos seguintes requisitos:

- Se você usar um usuário do IAM, sua função do Amazon SQS deve 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 do Lambda, adicione `okms:Decrypt` para a função de execução do Lambda.

Para obter mais informações, consulte [Visão geral do gerenciamento de acesso no Amazon SQS \(p. 116\)](#).

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

1. Abra o console do Amazon SQS em <https://console.aws.amazon.com/sqs/>.
2. No painel de navegação, escolha Queues.
3. No Filas, escolha a fila a ser configurada.
4. Na página da fila, escolha o Triggers do Lambda.
5. No Triggers do Lambda, escolha um gatilho do Lambda.

Se a lista não incluir o gatilho do Lambda que você precisa, escolha Configurar gatilho da função Lambda. Insira o nome de recurso da Amazon (ARN) da função do Lambda ou escolha um recurso existente. Em seguida, escolha Save (Salvar).

6. Escolha Save (Salvar). O console salva a configuração e exibe o Detalhes para a fila.

No Detalhes, o Triggers do Lambda exibe a função do Lambda e seu status. Demora aproximadamente 1 minuto para a função do Lambda se tornar associada à sua fila.

7. Para verificar os resultados da configuração, [Enviar uma mensagem para sua fila do \(p. 20\)](#) e, em seguida, visualizar a função do Lambda acionada no console do Lambda.

Gerenciando filas do Amazon SQS (console)

Depois de criar e configurar sua fila, você pode usar o console do Amazon SQS para enviar mensagens para a fila e recuperar da fila.

Tópicos

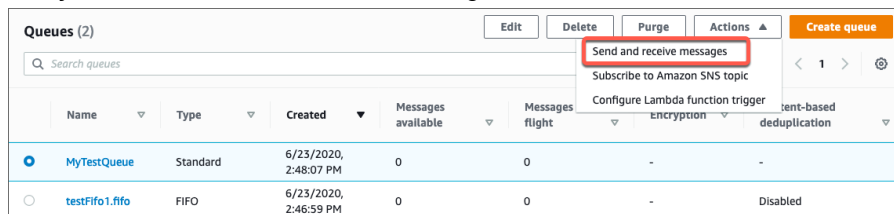
- [Enviar mensagens para uma fila \(console\) \(p. 20\)](#)
- [Enviar uma mensagem com atributos \(console\) \(p. 21\)](#)
- [Recebendo e excluindo mensagens \(console\) \(p. 22\)](#)
- [Limpar mensagens de uma fila do Amazon SQS \(console\) \(p. 22\)](#)
- [Excluir uma fila do Amazon SQS \(p. 23\)](#)
- [Confirmando que uma fila está vazia \(p. 23\)](#)

Enviar mensagens para uma fila (console)

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

Para enviar uma mensagem (console)

1. Abra o console do Amazon SQS em <https://console.aws.amazon.com/sqs/>.
2. No painel de navegação, escolha Queues.
3. No Filas, escolha uma fila.
4. No Ações, escolha Enviar e receber mensagens.

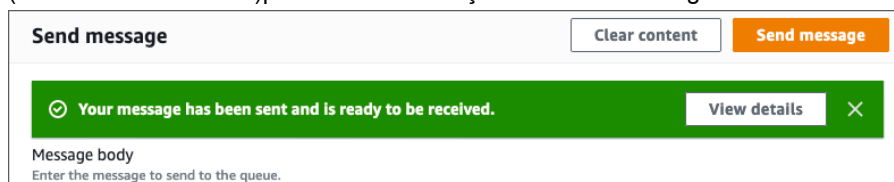


O console exibe o Enviar e receber mensagens.

5. No Corpo da mensagem, digite o texto da mensagem.
6. Para uma fila FIFO, insira um ID do grupo de mensagens. Para obter mais informações, consulte [Lógica da entrega FIFO \(p. 30\)](#).
7. (Opcional) Para uma fila FIFO, você pode inserir um ID de eliminação de duplicação de mensagem. Se você habilitou a eliminação de duplicação baseada em conteúdo para a fila, o ID da eliminação de duplicação da mensagem não será necessário. Para obter mais informações, consulte [Lógica da entrega FIFO \(p. 30\)](#).
8. (Opcional) Para uma fila padrão, você pode inserir um valor para Retardo da entrega e escolha as unidades. Por exemplo, digite 60 e escolha segundos. As filas FIFO não suportam temporizadores em mensagens individuais. Para obter mais informações, consulte [Temporizadores de mensagem do Amazon SQS \(p. 52\)](#).

9. Escolha Send Message (Enviar mensagem).

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



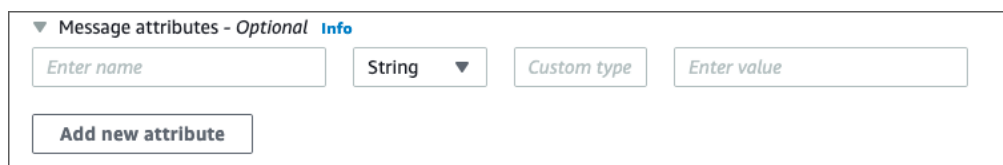
The screenshot shows the 'Send message' section of the Amazon SQS console. At the top, there are buttons for 'Clear content' and 'Send message'. Below this, a green banner displays a success message: 'Your message has been sent and is ready to be received.' with a 'View details' button and a close icon. Underneath the banner, the 'Message body' section is visible, with a prompt to 'Enter the message to send to the queue.'

Enviar uma mensagem com atributos (console)

Para filas padrão e FIFO, você pode incluir metadados estruturados (como carimbos de data e hora, dados geoespaciais, assinaturas e identificadores) com mensagens. Para obter mais informações, consulte [Atributos de mensagem do Amazon SQS \(p. 35\)](#).

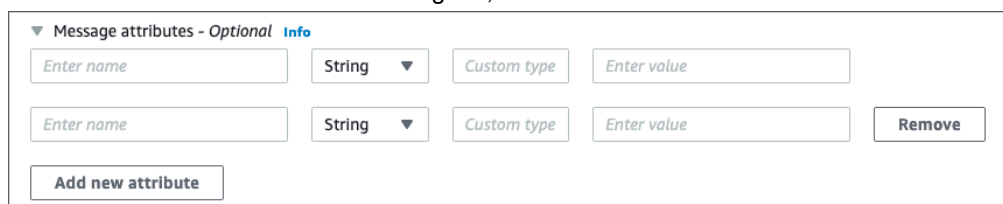
Para enviar uma mensagem com atributos a uma fila (console)

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



The screenshot shows the 'Message attributes - Optional' section in the Amazon SQS console. It features a form with fields for 'Enter name', a dropdown for 'String' (with a custom type option), and 'Enter value'. There is an 'Add new attribute' button at the bottom.

6. Para adicionar outro atributo de mensagem., escolha Adicionar novo atributo.



This screenshot shows the 'Message attributes' section after adding a second attribute. It displays two attribute rows, each with 'Enter name', a type dropdown, a custom type field, and a value field. A 'Remove' button is present next to the second attribute. An 'Add new attribute' button is at the bottom.

7. Você pode modificar os valores do atributo a qualquer momento antes de enviar a mensagem.
8. Para excluir um atributo, selecione Remove. Para excluir o primeiro atributo, feche o Atributos de mensagem.
9. Ao concluir a adição de atributos à mensagem, selecione enviar mensagem. A mensagem é enviada e o console exibe uma mensagem de sucesso. Para exibir informações sobre os atributos de mensagem da mensagem enviada, escolha View details (Visualizar os detalhes). Selecione Concluído Para fechar o Detalhes da mensagem Caixa de diálogo.

Recebendo e excluindo mensagens (console)

Depois de enviar mensagens para uma fila, você pode recebê-las e excluí-las. Quando você solicita mensagens de uma fila, não poderá especificar quais mensagens recuperar. Em vez disso, você especifica o número de mensagens máximo (até 10) que deseja recuperar.

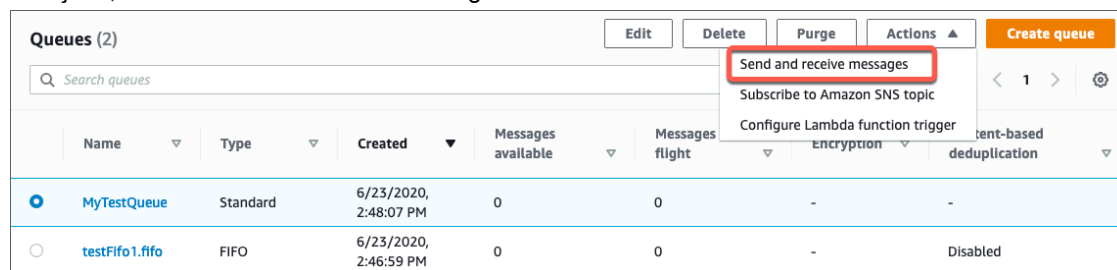
Note

Como o Amazon SQS é um sistema distribuído, uma fila com poucas mensagens pode exibir uma resposta vazia a uma solicitação de recebimento. Nesse caso, execute novamente a solicitação para receber sua mensagem. Dependendo das necessidades do seu aplicativo, você poderá precisar usar [sondagem curta ou longa do \(p. 39\)](#) para receber mensagens.

O Amazon SQS não exclui uma mensagem automaticamente após recuperá-la para você, caso não receba a mensagem com êxito (por exemplo, se os consumidores falharem ou você perder conectividade). Para excluir uma mensagem, você deverá enviar uma solicitação separada que reconheça que recebeu e processou com êxito. Observe que você deve receber uma mensagem antes de excluí-la.

Para receber e excluir uma mensagem (console)

1. Abra o console do Amazon SQS em <https://console.aws.amazon.com/sqs/>.
2. No painel de navegação, escolha Queues.
3. No Filas, escolha uma fila.
4. No Ações, escolha Enviar e receber mensagens.



O console exibe o Enviar e receber mensagens.

5. Selecione Sondagem de mensagens.

O Amazon SQS começa a pesquisar mensagens na fila. A barra de progresso no lado direito do Receber mensagens exibe a duração da sondagem.

O Mensagens exibe uma lista das mensagens recebidas. Para cada mensagem, a lista exibe o ID da mensagem, Data de envio, Tamanho e Contagem de recebimento.

6. Para excluir mensagens, escolha as mensagens que deseja excluir e escolha Excluir.
7. No Excluir mensagens, escolha Excluir.

Limpar mensagens de uma fila do Amazon SQS (console)

Se não quiser excluir uma fila do Amazon SQS, mas precisar excluir todas as mensagens dela, limpe-a. 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 quaisquer mensagens excluídas.

Para expurgar 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. NoFilas, escolha a fila a ser expurgado.
4. SelecioneLimpar.
5. NoLimpar fila, confirme a Limpar inserindopurgee escolhendoLimpar.

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

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 está vazia \(p. 23\)](#).

Você pode excluir uma fila, mesmo quando ela não estiver vazia. Para excluir as mensagens em uma fila, mas não a própria fila, [Limpar a fila \(p. 22\)](#).

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. NoFilasPara excluir a fila.
4. Escolha Delete (Excluir).
5. NoExcluir fila, confirme a exclusão inserindodelete.
6. Escolha Delete (Excluir).

Para excluir uma fila (AWS CLI AWS API)

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

- AWS CLI: [aws sqs delete-queue](#)
- AWS API: [DeleteQueue](#)

Confirmando que uma fila está vazia

Na maioria dos casos, você pode usar [sondagem longa \(p. 40\)](#) 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 especificou um valor baixo paraTempo de espera da mensagemQuando você criou a fila do. Esta seção descreve como confirmar se uma fila está vazia.

Para confirmar se uma fila está vazia (console)

1. Impedir 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. No Filas, escolha uma fila do.
5. Escolha a guia Monitoring (Monitoramento).
6. No canto superior direito dos painéis de monitoramento, escolha a seta para baixo ao lado do símbolo Atualizar. No menu suspenso, escolha Atualização automática. Deixe o Atualização do intervalo a 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 0 Por vários minutos, a fila está vazia.

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

1. Impedir o envio de mensagens por todos os produtores.
2. Execute repetidamente um dos seguintes comandos:
 - CLI DA AWS: `get-queue-attributes`
 - API DA AWS: `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 confiar nas métricas do Amazon CloudWatch, certifique-se de ver vários pontos de dados zero consecutivos antes de considerar essa fila vazia. Para obter mais informações sobre métricas do CloudWatch, consulte [Métricas CloudWatch Amazon SQS \(p. 151\)](#).

Como o Amazon SQS

Esta seção descreve os tipos de filas do Amazon SQS e suas propriedades básicas. Descreve também os identificadores de filas e mensagens e vários fluxos de trabalho de gerenciamento de filas e mensagens.

Tópicos

- [Arquitetura básica do Amazon SQS \(p. 25\)](#)
- [Filas padrão do Amazon SQS \(p. 28\)](#)
- [Filas FIFO do Amazon SQS \(p. 28\)](#)
- [Identificadores de filas e mensagens do Amazon SQS \(p. 33\)](#)
- [Metadados de mensagens \(p. 35\)](#)
- [Recursos necessários para processar mensagens do Amazon SQS \(p. 38\)](#)
- [Listar filas \(p. 38\)](#)
- [Tag de alocação de custos do Amazon SQS \(p. 39\)](#)
- [Sondagem curta e longa do Amazon SQS \(p. 39\)](#)
- [Filas de mensagens mortas do Amazon SQS \(p. 41\)](#)
- [Tempo limite de visibilidade do Amazon SQS \(p. 44\)](#)
- [Filas de atraso do Amazon SQS \(p. 47\)](#)
- [Filas temporárias do Amazon SQS \(p. 48\)](#)
- [Temporizadores de mensagem do Amazon SQS \(p. 52\)](#)

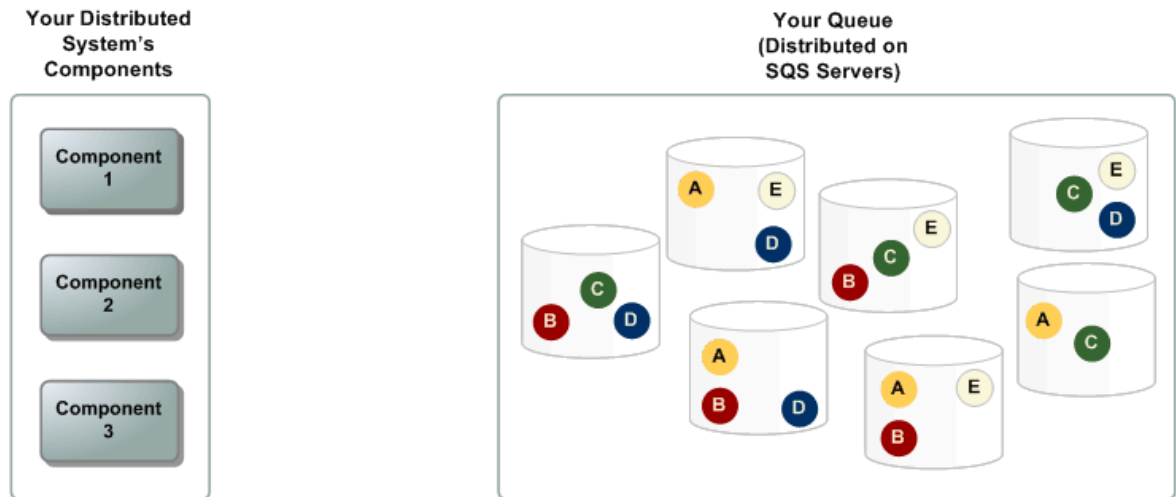
Arquitetura básica do Amazon SQS

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

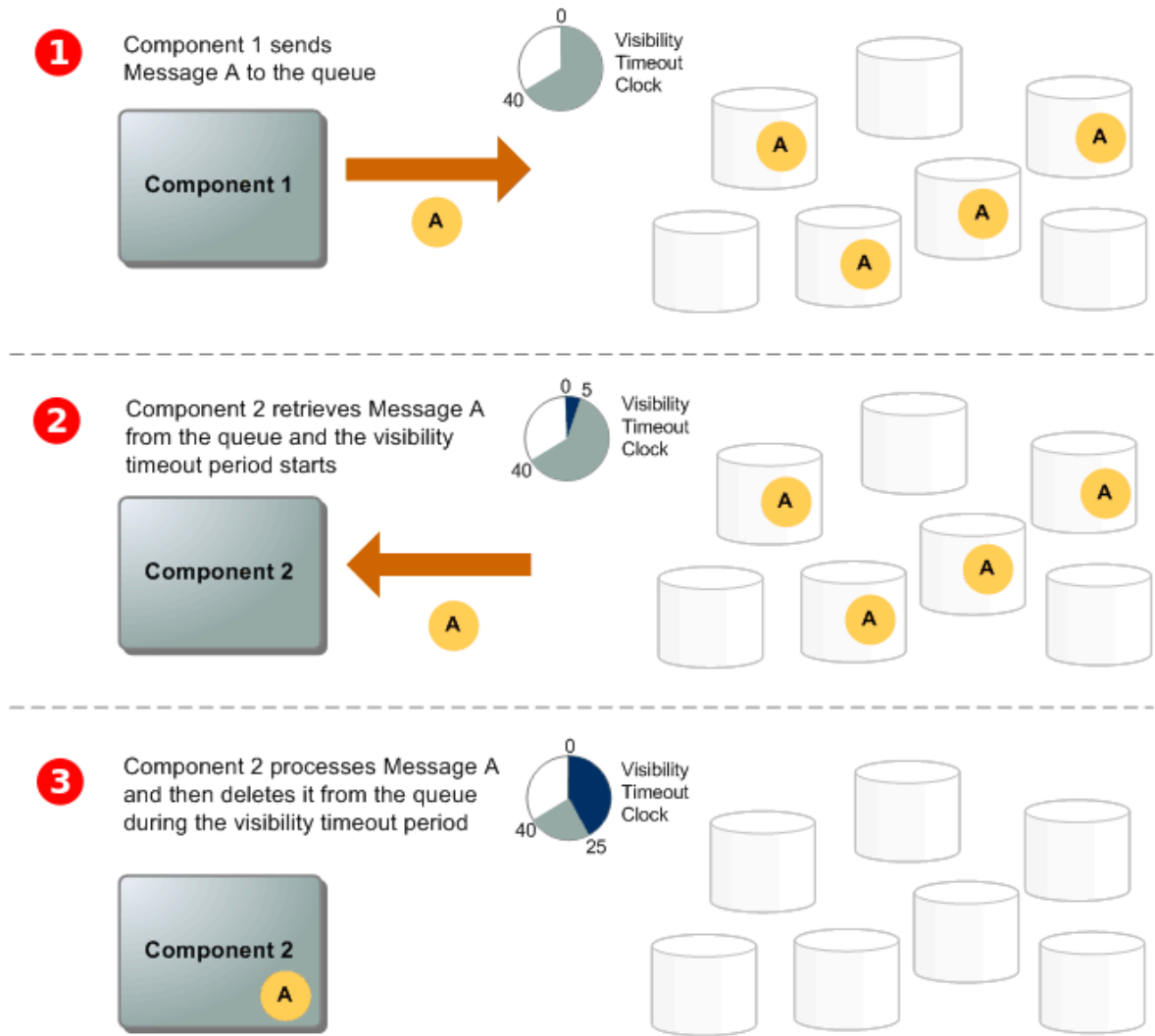
Há três componentes principais em um sistema de mensagens distribuído: os componentes do sistema distribuído, a fila (distribuída em servidores do Amazon SQS) e as mensagens na fila.

No cenário a seguir, seu sistema tem vários produtores do (componentes que enviam mensagens para a fila) e consumidor (componentes que recebem mensagens da fila). A fila (que contém as mensagens de 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 consome 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](#) (p. 44).

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 estão na fila há mais tempo do 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](#)

Filas padrão do Amazon SQS

Ofertas do Amazon SQS 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 mensagens pelo menos uma vez. No entanto, às vezes (devido à 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 geralmente garante a entrega das mensagens na mesma ordem em que foram enviadas.

Para obter informações sobre como criar e configurar filas usando o console do Amazon SQS, consulte [Criação de uma fila do Amazon SQS \(console\)](#) (p. 12). Para exemplos de Java, consulte [Exemplos de SDK Java do Amazon SQS](#) (p. 60).

É possível utilizar filas de mensagem padrão em diversos cenários, contanto que o aplicativo possa processar mensagens que chegam mais de uma vez e fora de ordem, por exemplo:

- Desacoplar solicitações do usuário de trabalhos intensos em segundo plano— Permite aos usuários enviar mídia redimensionando-a ou codificando-a.
- Alocar tarefas a vários nós de trabalho— Processa uma alta quantidade de solicitações de validação de cartão de crédito.
- Mensagens em Batch para processamento futuro— Programe várias entradas para adicioná-las ao banco de dados.

Para cotas relacionadas a filas padrão, consulte [Cotas relacionadas a filas](#) (p. 99).

Para obter as melhores práticas para trabalhar com filas padrão, consulte [Recomendações para filas do Amazon SQS e FIFO](#) (p. 53).

Ordenação de mensagens

Uma fila padrão faz o possível para preservar a ordem das mensagens, mas mais de uma cópia de uma mensagem pode ser entregue fora de ordem. Se o seu sistema exigir que a ordem seja preservada, recomendamos usar um [Fila FIFO \(primeiro a entrar, primeiro a sair\)](#) (p. 28) Ou adicionar informações sobre o sequenciamento em cada mensagem para que você possa reordenar as mensagens quando elas são recebidas.

Entrega pelo menos uma vez

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

Filas FIFO do Amazon SQS

A alta taxa de transferência para filas FIFO do Amazon SQS está em versão de teste e sujeito a alterações. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO](#) (p. 11).

As filas FIFO têm todos os recursos do [fila padrão \(p. 28\)](#).

FIFO (primeiro a entrar, último a sair) As filas são projetadas para aprimorar a troca de mensagens entre aplicativos quando a ordem das operações e dos eventos é crítica ou quando duplicatas não podem ser toleradas. Exemplos de situações em que você pode usar filas FIFO incluem o seguinte:

- Para garantir que os comandos inseridos pelo usuário sejam executados na ordem correta.
- Para exibir o preço do produto correto enviando modificações de preço na ordem correta.
- Para evitar que um aluno se inscreva em um curso antes de criar uma conta.

As filas FIFO também fornecem processamento exatamente uma vez, mas são limitadas a um determinado número de transações por segundo (TPS):

- Se você usa o [agrupamento em lote \(p. 166\)](#), as filas FIFO oferecem suporte a até 3.000 transações por segundo, por método de API (`SendMessageBatch`, `ReceiveMessage` ou `DeleteMessageBatch`). As 3000 transações 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`).

Note

O nome de uma fila FIFO deve terminar com o `.fifo` Sufixo. O sufixo conta para a cota de nome de fila de 80 caracteres. Para determinar se uma fila é [FIFO \(p. 28\)](#), você pode verificar se o nome da fila termina com o sufixo.

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

Para obter informações sobre como criar e configurar filas usando o console do Amazon SQS, consulte [Criação de uma fila do Amazon SQS \(console\) \(p. 12\)](#). Para exemplos de Java, consulte [Exemplos de SDK Java do Amazon SQS \(p. 60\)](#).

Para cotas relacionadas a filas FIFO, consulte [Cotas relacionadas a filas \(p. 99\)](#).

Para obter as práticas recomendadas para trabalhar com filas FIFO, consulte [Recomendações adicionais para filas FIFO do Amazon SQS \(p. 56\)](#) e [Recomendações para filas do Amazon SQS e FIFO \(p. 53\)](#).

Para obter informações sobre a compatibilidade de clientes e serviços com filas FIFO, consulte [Compatibility \(p. 32\)](#).

Tópicos

- [Ordenação de mensagens \(p. 29\)](#)
- [Principais termos \(p. 30\)](#)
- [Lógica da entrega FIFO \(p. 30\)](#)
- [Processamento exatamente uma vez \(p. 31\)](#)
- [Mudança de uma fila Padrão para uma fila FIFO \(p. 32\)](#)
- [Compatibility \(p. 32\)](#)

Ordenação de mensagens

A fila FIFO aprimora e complementa o [fila padrão \(p. 28\)](#). Os recursos mais importantes desse tipo de fila são [FIFO \(First-In-First-Out\) \(p. 30\)](#) e [Processamento exatamente uma vez \(p. 31\)](#):

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

Além disso, as filas FIFO suportam Grupos de mensagens que permitem vários grupos de mensagens ordenadas em uma única fila. Não há cota para o número de grupos de mensagens dentro de uma fila FIFO.

Principais termos

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

ID de eliminação de duplicação de mensagem

O token usado para deduplicação de mensagens enviadas. Se uma mensagem com um ID de deduplicação de mensagens específico for enviada com êxito, todas as mensagens enviadas com o mesmo ID de deduplicação de mensagens serão aceitas com êxito, mas não serão entregues durante o intervalo de deduplicação de 5 minutos.

Note

A deduplicação de mensagens se aplica a uma fila inteira, não a grupos de mensagens individuais.

O Amazon SQS continua a controlar o ID de deduplicação da mensagem mesmo depois que a mensagem é recebida e excluída.

ID do grupo de mensagens

A marca 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 deduplicação do `ReceiveMessage` chamadas.

Número de sequência

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

Lógica da entrega FIFO

Os conceitos a seguir podem ajudar a entender melhor o envio e o recebimento de mensagens do 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 reconhecerá 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 do 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 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 diferente 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 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 grupos 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.

Note

É possível receber até 10 mensagens em uma única chamada usando o parâmetro de solicitação `MaxNumberOfMessages` da `ReceiveMessage` ação. Essas mensagens mantêm a ordem FIFO e podem ter o mesmo ID de grupo de mensagens. Portanto, 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 que o produtor ou o consumidor tente várias tentativas:

- Se o produtor detectar uma falha `SendMessage`, ele pode tentar enviar novamente quantas vezes for necessário, usando o mesmo ID de deduplicação de mensagens. Supondo que o produtor receba pelo menos uma confirmação antes que o intervalo de deduplicação expire, várias tentativas não afetam a ordenação das mensagens nem introduzem duplicatas.
- Se o consumidor detectar uma falha `ReceiveMessage`, ele pode tentar 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, várias tentativas não afetam a ordenação das mensagens.
- Quando você recebe uma mensagem com um ID de grupo de mensagens, não há mais mensagens para o mesmo ID de grupo de mensagens são retornadas, a menos que você exclua a mensagem ou ela se torne visível.

Processamento exatamente uma vez

Ao contrário das filas padrão, as filas FIFO não apresentam mensagens duplicadas. As filas FIFO ajudam a evitar o envio de duplicações para uma fila. Se você tentar novamente o `SendMessage` ação em um 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 faz com que o Amazon SQS use um hash SHA-256 para gerar o ID de eliminação de duplicação de mensagem usando o corpo da mensagem, mas não os atributos da mensagem. Para obter mais informações, consulte a documentação

do [CreateQueue](#), [GetQueueAttributes](#), e [SetQueueAttributes](#) Ações em 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 do [SendMessage](#), [SendMessageBatch](#), e [ReceiveMessage](#) Ações em Referência da API do Amazon Simple Queue Service.

Mudança de uma fila Padrão para uma fila FIFO

Se você tiver um aplicativo que usa filas padrão e quiser aproveitar os recursos de ordenação ou processamento exatamente uma vez de filas FIFO, precisará configurar a fila e seu aplicativo corretamente.

Note

Não é possível converter uma fila padrão em uma fila FIFO. Para fazer a mudança, é necessário criar uma nova fila FIFO para o aplicativo ou excluir uma fila padrão atual e recriá-la como fila FIFO.

Para se certificar de que o aplicativo funciona corretamente com uma fila FIFO, use a seguinte lista de verificação:

- Se você usa o [agrupamento em lote](#) (p. 166), as filas FIFO oferecem suporte a até 3.000 transações por segundo, por método de API ([SendMessageBatch](#), [ReceiveMessage](#) ou [DeleteMessageBatch](#)). As 3000 transações 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](#)).
- 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.
- Cada mensagem enviada a uma fila FIFO exige um ID de grupo de mensagens. Se você não precisar de vários grupos de mensagens ordenadas, especifique o mesmo ID de grupo para todas as suas mensagens.
- 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 o [Referência da API do Amazon Simple Queue Service](#).

Compatibility

Cientes

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

Serviços

Se seu aplicativo usar vários serviços da AWS ou uma combinação de serviços da AWS e externos, é importante compreender qual funcionalidade de serviço não oferece suporte a filas FIFO.

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

Os seguintes recursos dos serviços da AWS não são compatíveis com filas FIFO:

- [Notificações de eventos do Amazon S3](#)
- [Auto Scaling ciclo de vida](#)
- [Ações de regras do AWS IoT](#)
- [Filas do AWS Lambda](#)

Para obter informações sobre a compatibilidade de outros serviços com filas FIFO, consulte a documentação do serviço.

Identificadores de filas e mensagens do Amazon SQS

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

Tópicos

- [Identificadores para filas padrão do Amazon SQS e FIFO \(p. 33\)](#)
- [Identificadores adicionais para filas FIFO do Amazon SQS \(p. 34\)](#)

Identificadores para filas padrão do Amazon SQS e FIFO

Para obter mais informações sobre os identificadores a seguir, consulte o [Referência da API do Amazon Simple Queue 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 `.fifo` sufixo. O sufixo conta para a cota de nome de fila de 80 caracteres. Para determinar se uma fila é FIFO (p. 28), você pode verificar se o nome da fila termina com o sufixo.

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 obter mais informações, consulte [ListQueues](#).

ID de mensagem

Cada mensagem recebe um sistema atribuído ID de mensagem que o Amazon SQS retorna para você no [SendMessage](#) Resposta. Esse identificador é útil para identificar mensagens. No entanto, para excluir uma mensagem, você precisa do identificador de recebimento da mensagem. 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  
Lj1FjgXUv1uSjlGUPAWV66FU/WeR4mq2OKpEGYWbnLmpRCJVAyeMjeU5ZBdtcQ+QE  
auMZc8ZRv37sIW2iJKq3M9MFx1YvV11A2x/KSbkJ0=
```

Identificadores adicionais para filas FIFO do Amazon SQS

Para obter mais informações sobre os identificadores a seguir, consulte [Processamento exatamente uma vez \(p. 31\)](#) e a [Referência da API do Amazon Simple Queue Service](#).

ID de eliminação de duplicação de mensagem

O token usado para deduplicação de mensagens enviadas. Se uma mensagem com um ID de deduplicação de mensagens específico for enviada com êxito, todas as mensagens enviadas com o mesmo ID de deduplicação de mensagens serão aceitas com êxito, mas não serão entregues durante o intervalo de deduplicação de 5 minutos.

ID do grupo de mensagens

A marca 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.

Metadados de mensagens

É possível usar atributos de mensagem para associar metadados personalizados a mensagens do Amazon SQS para seus aplicativos. 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 \(p. 35\)](#)
- [Amazon SQS atributos do sistema de mensagens do Amazon \(p. 37\)](#)

Atributos de mensagem do Amazon SQS

O Amazon SQS permite incluir metadados estruturados (como carimbos de data e hora, dados geoespaciais, assinaturas e identificadores) com mensagens do usando Atributos de mensagens do. 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 \(console\) \(p. 21\)](#).

Note

Não confunda atributos de mensagem com Atributos do sistema de mensagens do: Enquanto é possível usar atributos de mensagem para associar metadados personalizados a mensagens do Amazon SQS para seus aplicativos, é possível usar [Atributos do sistema de mensagens do \(p. 37\)](#) para armazenar metadados para outros serviços da AWS, como o AWS X-Ray.

Tópicos

- [Componentes de atributos de mensagem \(p. 35\)](#)
- [Tipos de dados de atributos de mensagem \(p. 36\)](#)
- [Cálculo do resumo de mensagens MD5 para atributos de mensagem \(p. 36\)](#)

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 período (`.`). 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 com suporte incluem `string`, `Number`, `eBinary`. 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](#) no 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— 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 mensagem indicam ao Amazon SQS, como tratar valores de atributos de mensagem correspondentes. Por exemplo, se o tipo for `Number`, o Amazon SQS valida valores numéricos.

O Amazon SQS oferece suporte aos tipos de dados lógicos `String`, `Number`, e `Binary` com rótulos de tipos de dados personalizados opcionais com o formato `.custom-data-type`

- `String:String` Os atributos podem armazenar texto Unicode usando quaisquer caracteres XML válidos.
- `telefone:Number` Os atributos 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 quaisquer dados binários, como dados compactados, dados criptografados ou imagens.
- `Personalizar`— para criar um tipo de dados personalizado, acrescente um rótulo de tipo personalizado a qualquer tipo de dados. 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 acrescentados.

O rótulo de tipo personalizado tem as mesmas restrições que o corpo da mensagem.

Cálculo do resumo de mensagens MD5 para atributos de mensagem

Ao usar o AWS SDK for Java, você pode ignorar esta seção. O `MessageMD5ChecksumHandler` do SDK for Java oferece suporte a resumos de mensagens MD5 para atributos de mensagens do Amazon SQS.

Ao usar a API de consulta ou um dos SDKs da AWS que não oferece suporte ao resumo de mensagens MD5 para atributos de mensagens do Amazon SQS, use as seguintes diretrizes para executar o cálculo do resumo de mensagens MD5.

Note

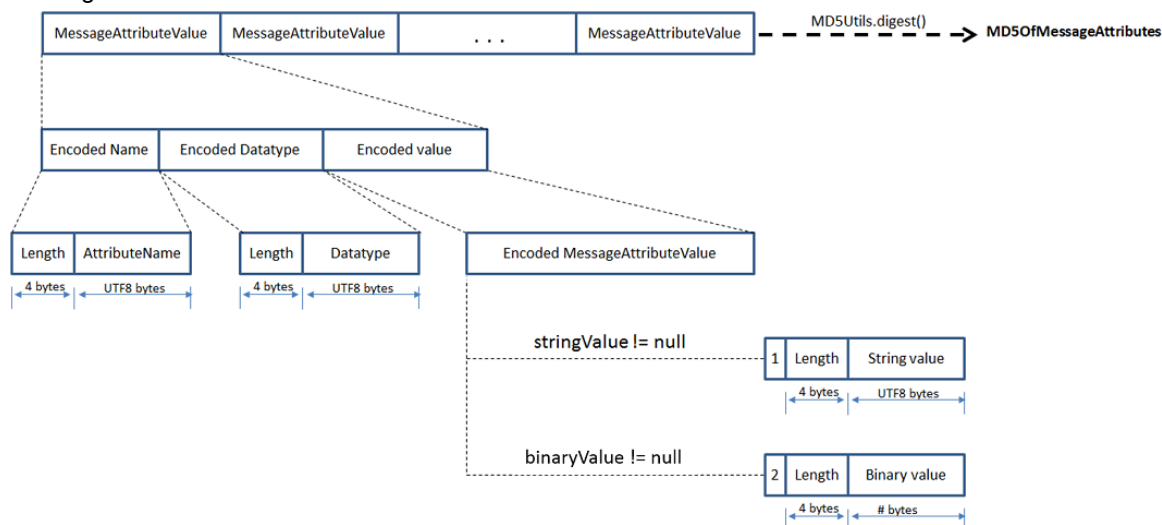
Sempre inclua sufixos de tipo de dados personalizados no cálculo de mensagem MD5.

Overview

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.

Amazon SQS atributos do sistema de mensagens do Amazon

Enquanto é possível usar [Atributos de mensagens do \(p. 35\)](#) Para associar metadados personalizados a mensagens do Amazon SQS para seus aplicativos, você pode usar [Atributos do sistema de mensagens](#) para armazenar metadados para outros serviços da AWS, como o AWS X-Ray. Para obter mais informações, consulte o `MessageSystemAttribute` parâmetro de solicitação do `SendMessage` `SendMessageBatch` Ações da API, o `AWSTraceHeader` Atributo do `ReceiveMessage`

a ação da API `MessageSystemAttributeValue` Tipo de dados noReferência 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` seu valor deve ser uma string de cabeçalho de rastreamento do AWS X-Ray 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 ajudá-lo a estimar os recursos necessários para processar mensagens na fila, o Amazon SQS pode determinar o número aproximado 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 \(p. 44\)](#).

Note

Para filas padrão, o resultado é aproximado devido à arquitetura distribuída do Amazon SQS. Na maioria dos casos, a contagem deve estar próxima do número 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.	<code>ApproximateNumberOfMessages</code>
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.	<code>ApproximateNumberOfMessagesDelayed</code>
Obter o número de mensagens que estão em processamento. 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.	<code>ApproximateNumberOfMessagesNotVisible</code>

Listar filas

Os métodos de API suportam controles opcionais de paginação. Por padrão, esses métodos de API retornam até 1000 filas na mensagem de resposta. É possível definir a propriedade `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 `MaxResults` a resposta incluir um máximo de 1.000 resultados e `NextToken` na resposta é nulo.

Se você definir `MaxResults` a resposta incluir 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, `NextToken` na resposta é nulo.

Tag de alocação de custos do Amazon SQS

Para organizar e identificar suas filas do Amazon SQS para alocação de custos, você pode adicionar metadados tags que identificam um objetivo da fila, proprietário ou ambiente. —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”](#) (p. 17)

Você pode usar tags de alocação de custos para organizar sua conta AWS e refletir sua própria estrutura de custos. Para isso, cadastre-se para obter sua conta da AWS com os valores de chave de tags incluídos. Para obter mais informações, consulte [Como configurar um relatório mensal de alocação de custos no AWS Billing and Cost Management](#) Guia do usuário.

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.
- Tags não têm significado semântico. O Amazon SQS interpreta tags como sequências de caracteres.
- As tags diferenciam 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 5 TPS por conta da AWS. Se o seu aplicativo exigir uma taxa de transferência mais alta, [Envie uma solicitação](#).

Para obter uma lista completa das restrições de tags do, consulte [Cotas relacionadas a filas](#) (p. 99).

Sondagem curta e longa do Amazon SQS

O Amazon SQS oferece a sondagem curta e a sondagem longa para receber mensagens de uma fila. Por padrão, as filas usam a sondagem curta.

comsondagem curta, o`ReceiveMessage`Solicitar consultas apenas um subconjunto dos servidores (com base em uma distribuição aleatória ponderada) para localizar mensagens que estejam disponíveis para incluir na resposta. O Amazon SQS envia a resposta imediatamente, mesmo que a consulta não tenha encontrado nenhuma mensagem.

comsondagem longa, o`ReceiveMessage`consulta todos os servidores para mensagens. O Amazon SQS envia uma resposta depois de coletar pelo menos uma mensagem disponível, até o número máximo de mensagens especificado na solicitação. O Amazon SQS envia uma resposta vazia somente se o tempo de espera de sondagem expirar.

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

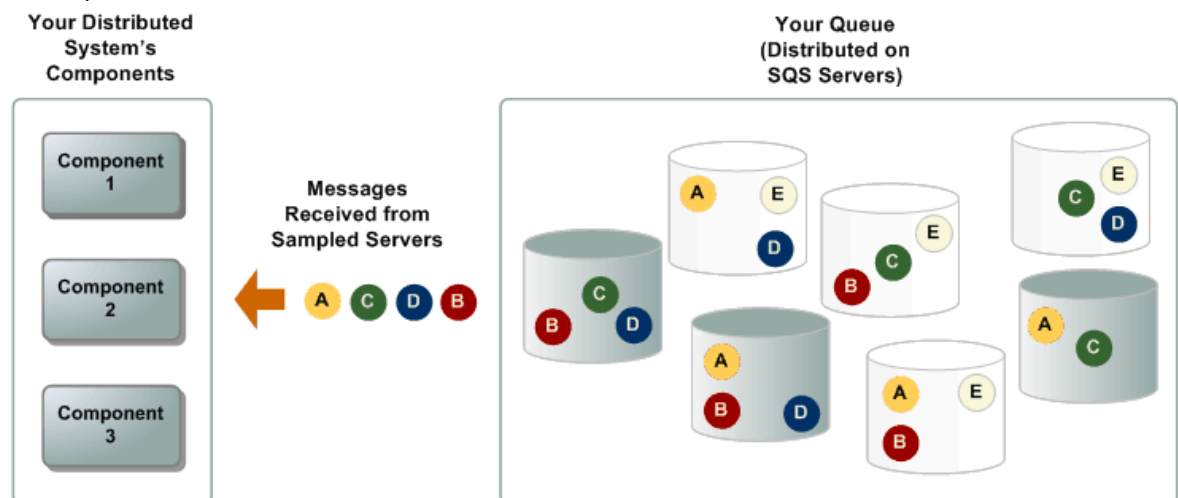
Tópicos

- [Consumo de mensagens usando sondagem curta \(p. 40\)](#)
- [Consumo de mensagens usando sondagem longa \(p. 40\)](#)
- [Diferenças entre as sondagens longa e curta \(p. 41\)](#)

Consumo de mensagens usando sondagem curta

Quando você consome as mensagens de uma fila usando a sondagem curta, o Amazon SQS tira amostras de um subconjunto de seus servidores (com base em uma distribuição aleatória ponderada) e retorna mensagens apenas 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 analisará todos os seus servidores, e você receberá todas as suas mensagens.

O diagrama a seguir mostra o comportamento de sondagem breve de mensagens retornadas de uma fila padrão depois que um dos componentes do sistema faz uma solicitação de recebimento. O Amazon SQS tira amostras de 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 sondagem longa

Quando o tempo de espera para o`ReceiveMessage`A ação da API é maior que 0,sondagem longaestá 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 um [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 [aConfigurando parâmetros de fila \(console\) \(p. 14\)](#). Para ver as melhores práticas, consulte [Configuração da sondagem longa \(p. 54\)](#).

A sondagem longa oferece os seguintes benefícios:

- Reduza respostas vazias permitindo que o Amazon SQS espere até que uma mensagem esteja disponível em uma fila para 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 contém mensagens, especialmente se você especificar um valor baixo para o `ReceiveMessageWaitTimeSeconds` parâmetro.
- Reduza respostas vazias falsas consultando todos os servidores do Amazon SQS, em vez de um subconjunto deles.
- Retornar mensagens assim que se tornam disponíveis.

Para obter informações sobre como confirmar que uma fila está vazia, consulte [Confirmando que uma fila está vazia \(p. 23\)](#).

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:

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

Filas de mensagens mortas do Amazon SQS

O Amazon SQS oferece suporte a Filas de mensagens mortas do, quais outras filas (filas de origem) podem usar como destinos para mensagens que não podem ser processadas (consumidas) com êxito. As dead letter queues são úteis para depurar seu aplicativo ou sistema de mensagens, pois permitem que você isole mensagens problemáticas para determinar por que seu processamento não obteve êxito. Para obter informações sobre como criar uma fila e configurar uma dead-letter queue para essa fila usando o console do Amazon SQS, consulte [Configurar uma dead letter queue \(console\) \(p. 16\)](#).

Important

O Amazon SQS não cria a dead-letter queue automaticamente. Você deve primeiro criar a fila antes de usá-la como uma dead-letter queue.

Tópicos

- [Como as filas de mensagens mortas funcionam? \(p. 42\)](#)
- [Quais são os benefícios das filas de mensagens mortas? \(p. 42\)](#)
- [Como os diferentes tipos de fila lidam com falhas de mensagens? \(p. 43\)](#)
- [Quando devo usar uma fila de mensagens mortas? \(p. 43\)](#)
- [Solução de problemas de filas de mensagens mortas \(p. 44\)](#)

Como as filas de mensagens mortas funcionam?

Algumas vezes, as mensagens não podem ser processadas devido a uma variedade de possíveis problemas, como condições de erro dos aplicativos de produtores/consumidores ou uma mudança de estado inesperada que causa um problema com o código do seu aplicativo. Por exemplo, se um usuário fizer um pedido na web com um ID de produto específico, mas esse ID do produto for excluído, o código da loja da web falhará e exibirá um erro. Além disso, a mensagem com a solicitação do pedido será enviada para uma dead-letter queue.

Ocasionalmente, os produtores e os consumidores podem não conseguir interpretar aspectos do protocolo que eles usam para se comunicar, causando a perda ou corrupção de mensagens. Além disso, os erros de hardware do consumidor podem corromper o conteúdo da mensagem.

Política de redirecionamentoEspecifica o fila de origem, oFila de mensagens mortasE as condições sob as quais o Amazon SQS moverá mensagens da primeira para a segunda, caso o consumidor da fila de origem não consiga processar uma mensagem um número especificado de vezes. Quando oReceiveCountpara uma mensagem excede omaxReceiveCountPara uma fila, o Amazon SQS moverá a mensagem para uma dead-letter queue (com o ID da mensagem original). Por exemplo, se a fila de origem tiver uma política de redirecionamento commaxReceiveCountDefinir como 5, e o consumidor da fila de origem receber uma mensagem 6 vezes sem excluí-la, o Amazon SQS moverá a mensagem para a dead-letter queue.

Para especificar uma dead-letter queue, você pode usar o console ou o AWS SDK for Java. Você deve fazer isso para cada fila que envia mensagens para uma dead letter queue. Várias filas do mesmo tipo podem se dirigir a uma única fila de mensagens mortas. Para obter mais informações, consulte [Configurar uma dead letter queue \(console\)](#) (p. 16) e o atributo RedrivePolicy da ação da `CreateQueue` ou `SetQueueAttributes`.

Important

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.

Você deve usar a mesma conta da AWS para criar a dead letter queue e outras filas que enviam mensagens para a dead letter queue. Além disso, as dead letter queues devem residir na mesma região que as outras filas que usam a dead letter queue. Por exemplo, se você criar uma fila na região Leste dos EUA (Ohio) e desejar usar uma dead-letter queue com essa fila, a segunda fila também deverá estar na região Leste dos EUA (Ohio).

A expiração de uma mensagem é sempre baseada em seu carimbo de data e hora de enfileiramento original. Quando uma mensagem é movida para uma dead-letter queue, o carimbo de data e hora do enfileiramento não é alterado. O `ApproximateAgeOfOldestMessage` indica quando a mensagem foi movida para a fila de letras mortas,nãoQuando a mensagem foi originalmente enviada. Por exemplo, suponha que uma mensagem gasta 1 dia na fila original antes de ser movida para uma fila de letras mortas. Se o período de retenção da fila de letras mortas for de 4 dias, a mensagem é excluída da fila de letras mortas após 3 dias e a propriedade `ApproximateAgeOfOldestMessage` é de três dias. Portanto, é uma prática recomendada definir sempre o período de retenção de uma fila de letras mortas para ser maior do que o período de retenção da fila original.

Quais são os benefícios das filas de mensagens mortas?

A principal tarefa de uma dead letter queue é lidar com falhas de mensagem. Uma dead letter queue permite que você separe e isole mensagens que não podem ser processadas corretamente para determinar por que seu processamento não obteve êxito. Configurar uma dead letter queue permite que você faça o seguinte:

- Configurar um alarme para todas as mensagens entregues para uma dead letter queue.
- Examinar logs para encontrar as exceções que podem ter causado a entrega de mensagens a uma dead letter queue.
- Analisar o conteúdo de mensagens entregues a uma dead letter queue para diagnosticar problemas de software ou de hardware do produtor ou consumidor.
- Determinar se você concedeu ao consumidor tempo suficiente para processar mensagens.

Como os diferentes tipos de fila lidam com falhas de mensagens?

Filas padrão

[Filas padrão \(p. 28\)](#) Mantenha o processamento de mensagens até a expiração do período de retenção. Esse processamento contínuo de mensagens minimiza as chances de ter sua fila bloqueada por mensagens que não podem ser processadas. O processamento contínuo de mensagens também oferece uma recuperação mais rápida da fila.

Em um sistema que processa milhares de mensagens, ter um grande número de mensagens que o consumidor repetidamente falha em confirmar e excluir pode aumentar os custos e colocar uma carga extra no hardware. Em vez de tentar processar mensagens com falha até que elas expirem, é melhor movê-las para uma dead letter queue após algumas tentativas de processamento.

Note

As filas padrão permitem um grande número de mensagens em trânsito. Se a maioria de suas mensagens não puder ser consumida e não for enviada para uma dead letter queue, sua taxa de processamento de mensagens válidas poderá diminuir. Dessa forma, para manter a eficiência da fila, certifique-se de que o aplicativo lide com o processamento de mensagens corretamente.

Filas FIFO

[Filas FIFO \(p. 28\)](#) Fornecem o processamento exatamente uma vez consumindo mensagens de um grupo de mensagens em sequência. Assim, embora o consumidor possa continuar a recuperar mensagens ordenadas de um outro grupo de mensagens, o primeiro grupo de mensagens permanece indisponível até que a mensagem que está bloqueando a fila seja processada com êxito.

Note

As filas FIFO permitem um número menor de mensagens em trânsito. Assim, para impedir que sua fila FIFO seja bloqueada por uma mensagem, certifique-se de que seu aplicativo lida corretamente com o processamento de mensagens.

Quando devo usar uma fila de mensagens mortas?



Use dead-letter queues com filas padrão. Você deve tirar proveito das dead letter queues sempre que seus aplicativos não dependerem de ordenação. As dead letter queues podem ajudá-lo a solucionar as operações de transmissão de mensagens incorretas.

Note

Mesmo quando você usa dead letter queues, deve continuar a monitorar suas filas e tentar enviar mensagens que falham por razões transitórias.



Utilize as dead letter queues para diminuir o número de mensagens e reduzir a possibilidade de exposição de seu sistema a mensagens poison-pill (mensagens que podem ser recebidas, mas que não podem ser processadas).



Não use uma dead-letter queue com filas padrão quando quiser repetir a transmissão de uma mensagem indefinidamente. Por exemplo, não use uma dead letter queue se o programa precisar esperar que um processo dependente se torne ativo ou disponível.



Não use uma dead-letter queue com uma fila FIFO se não quiser quebrar a ordem exata de mensagens ou operações. Por exemplo, não use uma dead letter queue com instruções em uma Edit Decision List (EDL) para um pacote de edição de vídeo, onde a alteração da ordem de edições muda o contexto de edições subsequentes.

Solução de problemas de filas de mensagens mortas

Em alguns casos, as dead-letter queues do Amazon SQS podem não se comportar como esperado. Esta seção fornece uma visão geral dos problemas comuns e mostra como resolvê-los.

Visualizar mensagens usando o console pode fazer com que as mensagens sejam movidas para uma fila de mensagens mortas

O Amazon SQS considera a visualização de uma mensagem no console contra a política de redirecionamento da fila correspondente. Dessa forma, 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 dead-letter queue da fila correspondente.

Para ajustar esse comportamento, você pode executar uma das seguintes ações:

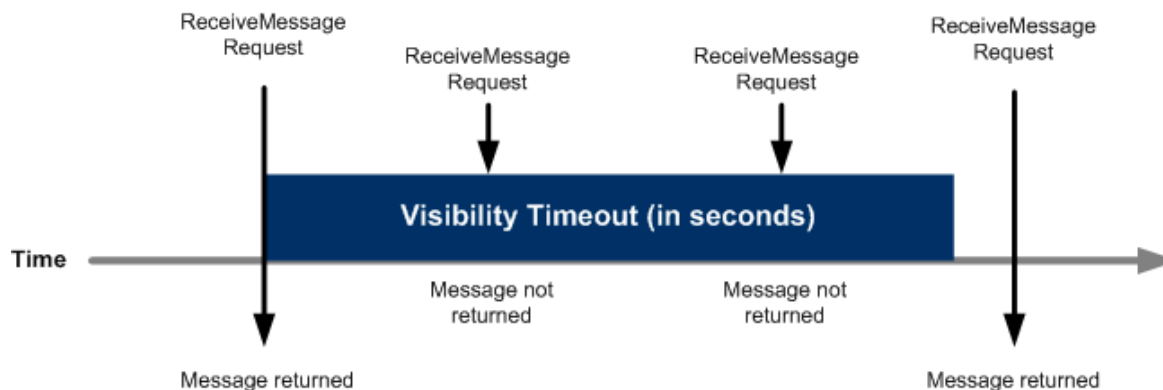
- Aumentar a definição de Maximum Receives da política de redirecionamento da fila correspondente.
- Evitar a visualização de mensagens da fila correspondente no console.

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. Assim, é possível que os valores de `NumberOfMessagesSent` e `NumberOfMessagesReceived` sejam diferentes.

Tempo limite de visibilidade do Amazon SQS

Quando um consumidor recebe e processa uma mensagem de uma fila, a mensagem 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, devido a um problema de conectividade ou devido a um problema no aplicativo consumidor). Desse modo, o consumidor deve excluir a mensagem da fila após o recebimento e processamento.



Imediatamente após o recebimento de uma mensagem, 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 outros consumidores recebam e processem a mensagem. O tempo limite de visibilidade padrão para uma mensagem é de 30 segundos. O mínimo é de 0 segundo. O máximo é de 12 horas. Para obter informações sobre como configurar o tempo limite de visibilidade para uma fila usando o console do, consulte [Configurando parâmetros de fila \(console\)](#) (p. 14).

Note

Para filas padrão, o tempo limite de visibilidade não é uma garantia contra o recebimento de uma mensagem duas vezes. Para obter mais informações, consulte [Entrega pelo menos uma vez](#) (p. 28).

As filas FIFO permitem que o produtor ou o consumidor tente várias tentativas:

- Se o produtor detectar uma falha `SendMessage`, ele pode tentar enviar novamente quantas vezes for necessário, usando o mesmo ID de deduplicação de mensagens. Supondo que o produtor receba pelo menos uma confirmação antes que o intervalo de deduplicação expire, várias tentativas não afetam a ordenação das mensagens nem introduzem duplicatas.
- Se o consumidor detectar uma falha `ReceiveMessage`, ele pode tentar 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, várias tentativas não afetam a ordenação das mensagens.
- Quando você recebe uma mensagem com um ID de grupo de mensagens, não há mais mensagens 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](#) (p. 45)
- [Definição do tempo limite de visibilidade](#) (p. 46)
- [Alteração do tempo limite de visibilidade de uma mensagem](#) (p. 46)
- [Término do tempo limite de visibilidade de uma mensagem](#) (p. 47)

Mensagens em trânsito

Uma mensagem do Amazon SQS tem três estados básicos:

1. Enviado para uma fila por um produtor.
2. Recebido da fila por um consumidor.
3. Excluído da fila.

Uma mensagem é considerada como armazenada depois de ser enviado para uma fila por um produtor, mas ainda não recebido 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 como em vias de ser recebido de uma fila por um consumidor, mas ainda não excluído 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 de bordo não estão relacionadas com um número limitado de mensagens armazenadas.

Para a maioria das filas padrão (dependendo do tráfego da fila e do acúmulo de mensagens), pode haver um máximo de aproximadamente 120.000 mensagens em processamento (recebidas de uma fila por um consumidor, mas ainda não excluídas da fila). Se você atingir essa cota ao usar o [Sondagem curta](#) (p. 40), o Amazon SQS retorna o erro `OverLimit`. Se você usar o [Sondagem longa](#) (p. 40), o Amazon SQS não retorna 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 um 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.

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](#) (p. 53).

Se não souber quanto tempo leva para processar uma mensagem, crie um [Heartbeat](#) para o seu processo de consumo: Especifique o tempo limite de visibilidade inicial (por exemplo, 2 minutos) e, em seguida, desde que o consumidor ainda trabalhe 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 o [ReceiveMessage](#) solicitação. Estender o tempo limite de visibilidade não redefine o período máximo de 12 horas. Se o cliente precisar de mais de 12 horas, considere usar [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 aplica-se apenas ao recebimento específico da mensagem. A ação `ChangeMessageVisibility` não afeta o tempo limite de recebimentos posteriores da mensagem ou 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 encerre 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

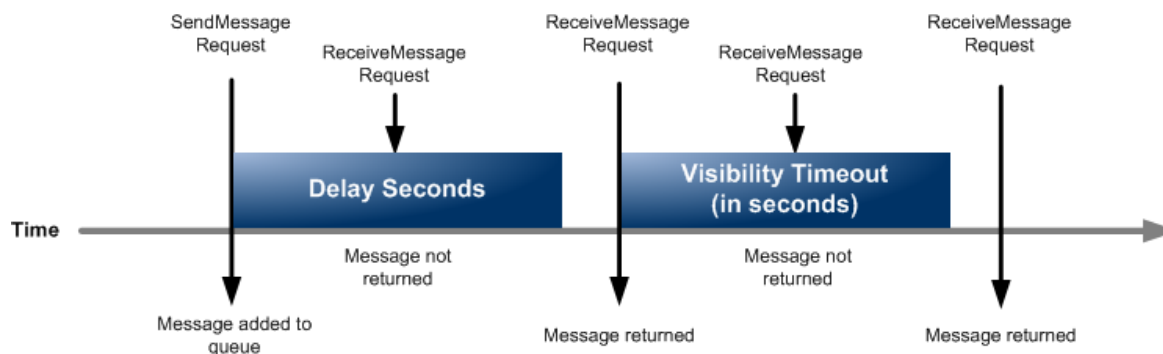
As filas de atraso permitem adiar a entrega de novas mensagens para uma fila por alguns segundos, por exemplo, quando o aplicativo do consumidor precisa de tempo adicional para processar mensagens. Se você criar uma fila de atraso, todas as mensagens enviadas para a fila permanecerão invisíveis para os consumidores durante o período de atraso do. O atraso padrão (mínimo) para uma fila é 0 segundos. O máximo é 15 minutos. Para obter informações sobre como configurar filas de atraso usando o console, consulte [Configurando parâmetros de fila \(console\)](#) (p. 14).

Note

Para filas padrão, a configuração de atraso por fila é não 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](#) (p. 44), 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 segundos de atraso em Mensagens individuais do, em vez de em uma fila inteira, use [Temporizadores de mensagens do](#) (p. 52) para permitir que o Amazon SQS use a propriedade `DelaySeconds` em vez do valor da fila de atraso `DelaySecondsValue`.

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 de resposta. Você pode usar o [Filas temporárias](#) para criar filas temporárias gerenciadas por aplicativos de alto débito, econômicas e econômicas.

O cliente mapeia várias Filas temporárias do—filas gerenciadas pelo aplicativo 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 do Amazon SQS estáticas do (normais). Isso significa que o código existente que envia e recebe mensagens pode enviar e receber mensagens de filas virtuais.

Tópicos

- [Filas virtuais](#) (p. 48)
- [Padrão de mensagens de resposta a solicitação \(filas virtuais\)](#) (p. 49)
- [Cenário de exemplo: Processamento de uma solicitação de login](#) (p. 50)
 - [No lado do cliente](#) (p. 50)
 - [No lado do servidor](#) (p. 51)
- [Limpeza das filas](#) (p. 51)

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 melhores práticas, consulte [Evitar reutilizar o mesmo ID de grupo de mensagens com filas virtuais](#) (p. 59).

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 obter mais informações, consulte [Cotas relacionadas a mensagens](#) (p. 100).

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.

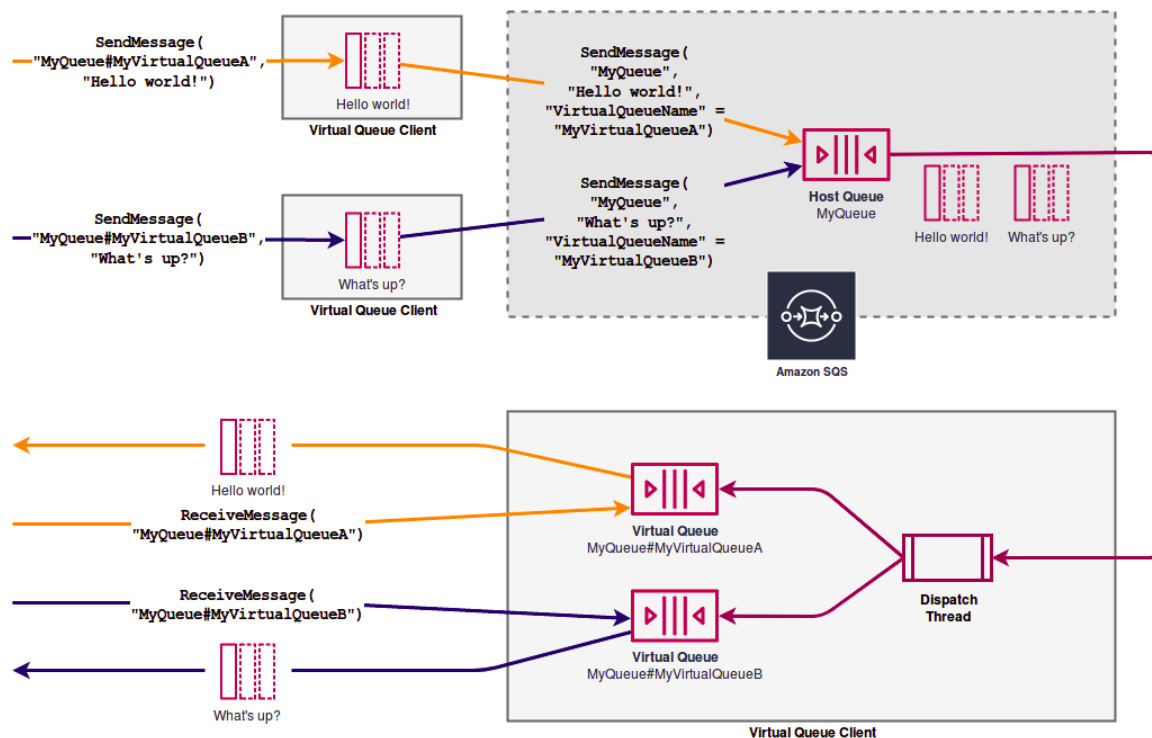
Enquanto o consumidor chama a ação de API `ReceiveMessage` em um URL de fila virtual, o `Temporary Queue Client` bloqueia a chamada localmente até o thread de fundo enviar uma mensagem para a fila virtual. (Esse processo é semelhante à pré-busca de mensagens no [Cliente assíncrono armazenado em buffer](#) (p. 167): 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` permite criar e excluir várias filas temporárias sem fazer chamadas de API do Amazon SQS. Para obter mais informações, consulte [Implementação de sistemas de resposta a solicitação](#) (p. 55).

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



Cenário de exemplo: Processamento de 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();

    private final 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();  
  
    private final AmazonSQS(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 o aplicativo não precisar mais do Temporary Queue Client, ele deverá chamar `shutdown()` Método do. 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 marcar 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 mensagem do Amazon SQS

Os temporizadores de mensagem do 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 aos consumidores nos primeiros na fila. O atraso padrão (mínimo) para uma mensagem é 0 segundos. O máximo é 15 minutos. Para obter informações sobre como enviar mensagens com temporizadores usando o console do, consulte [Enviar mensagens para uma fila \(console\) \(p. 20\)](#).

Note

As filas FIFO não suportam temporizadores em mensagens individuais.

Para definir um período de atraso em uma fila inteira, em vez de mensagens individuais, use [filas de atraso \(p. 47\)](#). Uma definição de temporizador de mensagem para uma mensagem individual substitui qualquer `DelaySeconds` em uma fila de atraso do Amazon SQS.

Melhores práticas do Amazon SQS

Essas práticas recomendadas podem ajudá-lo a obter o máximo do Amazon SQS.

Tópicos

- [Recomendações para filas do Amazon SQS e FIFO \(p. 53\)](#)
- [Recomendações adicionais para filas FIFO do Amazon SQS \(p. 56\)](#)

Recomendações para filas do Amazon SQS e FIFO

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

Tópicos

- [Trabalhando com mensagens do Amazon SQS \(p. 53\)](#)
- [Reduzir custos do Amazon SQS \(p. 55\)](#)
- [Movendo de uma fila padrão do Amazon SQS para uma fila FIFO \(p. 56\)](#)

Trabalhando 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 \(p. 53\)](#)
- [Lidar com erros de solicitação \(p. 54\)](#)
- [Configuração da sondagem longa \(p. 54\)](#)
- [Capturar mensagens problemáticas \(p. 55\)](#)
- [Configurar a retenção da fila de mensagens mortas \(p. 55\)](#)
- [Evitar o processamento inconsistente de mensagens \(p. 55\)](#)
- [Implementação de sistemas de resposta a solicitação \(p. 55\)](#)

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 você está alocando 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 [Configuração do tempo limite de visibilidade](#) (p. 46).

- Se não souber quanto tempo leva para processar uma mensagem, crie um `Heartbeat` para o seu processo de consumo: Especifique o tempo limite de visibilidade inicial (por exemplo, 2 minutos) e, em seguida, desde que o consumidor ainda trabalhe 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 o `ReceiveMessage` solicitação. Estender o tempo limite de visibilidade não redefine o período máximo de 12 horas. Se o cliente precisar de mais de 12 horas, considere usar as Funções da Etapa.

Lidar com erros de solicitação

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

- Se você usa um AWS SDK, já tem a lógica de repetição e recuo à sua disposição. Para obter mais informações, consulte [Repetições de erro e recuo exponencial na AWS](#) na Referência geral dos Amazon Web Services.
- Se você não usa os recursos do AWS SDK para repetição e recuo, faça uma pausa (por exemplo, 200 ms) antes de tentar novamente o `ReceiveMessage` após 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 para o `ReceiveMessage` A ação da API é maior que 0, 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 um `ReceiveMessage`) e respostas vazias falsas (quando mensagens estão disponíveis, mas não são incluídas em uma resposta). Para obter mais informações, consulte [Sondagem curta e longa do Amazon SQS](#) (p. 39).

Para 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 tiver configurado um AWS SDK com um tempo limite menor, poderá ser necessário modificar o cliente do Amazon SQS para permitir solicitações mais longas ou para usar um tempo limite menor para sondagem longa.
- 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, verifique se o tempo limite de resposta HTTP para o `ReceiveMessage` solicitações é maior do que o `waitTimeSeconds` parâmetro. Para obter mais informações, consulte [ReceiveMessage](#).

Capturar mensagens problemáticas

Para capturar todas as mensagens que não podem ser processadas e coletar métricas precisas do CloudWatch, configure um [Fila de mensagens mortas](#) (p. 41).

- 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).
- A inclusão de uma mensagem com conteúdo malicioso em uma fila pode distorcer o [ApproximateAgeOfOldestMessage](#) (p. 151) A métrica do CloudWatch, fornecendo uma idade incorreta da mensagem poison pill. Configurar uma dead letter queue ajuda a evitar alarmes falsos ao usar essa métrica.

Configurar a retenção da fila de mensagens mortas

A expiração 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 e hora do enqueue não é alterado. O `ApproximateAgeOfOldestMessage` métrica indica quando a mensagem foi movida para a fila de letras mortas. Quando a mensagem foi enviada originalmente. Por exemplo, suponha que uma mensagem gasta 1 dia na fila original antes de ser movida para uma fila de letras mortas. Se o período de retenção da fila de letras mortas for de 4 dias, a mensagem é excluída da fila de letras mortas após 3 dias e a propriedade `ApproximateAgeOfOldestMessage` é de três dias. Portanto, é uma prática recomendada definir sempre o período de retenção de uma fila de letras mortas para ser maior do que o período de retenção da fila original.

Evitar o processamento inconsistente de mensagens

Como o Amazon SQS é um sistema distribuído, é possível para um consumidor não receber uma mensagem mesmo quando o Amazon SQS marca a mensagem como entregue ao retornar com êxito de um `ReceiveMessage` Chamada de método de API. 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](#) (p. 41).

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 respostas na inicialização, por produtor e use um atributo de mensagem de ID de correlação para mapear respostas para 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 \(filas virtuais\)](#) (p. 49).

Reduzir 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 o [Ações da API de lote do Amazon SQS \(p. 166\)](#).
- Para combinar armazenamento em buffer do lado do cliente com o agrupamento de solicitações em lote, use a sondagem longa junto com o [cliente assíncrono em buffer \(p. 167\)](#) incluído com o AWS SDK para 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 se tornam disponíveis.
- Para reduzir o custo do uso do Amazon SQS e diminuir o número de recebimentos vazios para uma fila vazia (respostas para `receiveMessage` que não retornam nenhuma mensagem), habilite a sondagem longa. Para obter mais informações, consulte [Sondagem longa do Amazon SQS \(p. 39\)](#).
- 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 se a fila do Amazon SQS sondada estiver 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.

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

Se não estiver configurando `DelaySeconds` em cada mensagem, você pode mover para uma fila FIFO fornecendo um ID de grupo de mensagens para cada mensagem enviada.

Para obter mais informações, consulte [Mudança de uma fila Padrão para uma fila FIFO \(p. 32\)](#).

Recomendações adicionais para filas FIFO do Amazon SQS

O alto throughput para filas do Amazon SQS está em versão de teste e sujeito a alterações. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#).

As melhores práticas a seguir podem ajudar 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 o [.SendMessageSendMessageBatch](#) Ações no [Referência da API do Amazon Simple Queue Service](#).

Tópicos

- [Uso do ID de eliminação de duplicação de mensagens do Amazon SQS \(p. 57\)](#)
- [Uso do ID do grupo de mensagens do Amazon SQS \(p. 58\)](#)
- [Uso do ID de tentativa de solicitação de recebimento do Amazon SQS \(p. 59\)](#)

Uso do ID de eliminação de duplicação de mensagens do Amazon SQS

O ID de eliminação de duplicação de mensagens é o token usado para eliminação de duplicação de mensagens enviadas. Se uma mensagem com um ID de desduplicação de mensagens específico for enviada com êxito, todas as mensagens enviadas com o mesmo ID de desduplicação de mensagens serão aceitas com êxito, mas não serão entregues durante o intervalo de desduplicação de 5 minutos.

Note

A desduplicação de mensagens se aplica a uma fila inteira, não a grupos de mensagens individuais.

O Amazon SQS continua a controlar o ID de desduplicação 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 mensagem para cada mensagem nos seguintes cenários:

- Mensagens enviadas com corpos idênticos que o Amazon SQS deve tratar como exclusivas.
- Mensagens enviadas com conteúdo idêntico, mas com diferentes atributos que o Amazon SQS deve tratar como exclusivas.
- 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.
- 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 novamente enviar ou receber solicitações 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 sensível ao tempo. Ao projetar seu aplicativo, garanta que o produtor e o consumidor possam se recuperar em caso de interrupção do cliente ou da rede.

- O produtor deve estar ciente do intervalo de eliminação de duplicação da fila. O Amazon SQS tem um mínimo intervalo de eliminação de duplicação de 5 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](#) (p. 41).

Como trabalhar com tempos limite de visibilidade

Para um desempenho ideal, defina a propriedade [Tempo limite de visibilidade do](#) (p. 44) 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](#) (p. 40) ou [sondagem longa](#) (p. 39).

Uso do ID do grupo de mensagens do Amazon SQS

O ID do grupo de mensagens é a marca 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 ID de grupo de mensagens (por exemplo, os dados da sessão para vários usuários). Nesse cenário, vários consumidores podem processar a fila, mas os dados da 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 um máximo de 20.000 mensagens de bordo (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. Se a fila tiver um grande backlog de 20.000 ou mais mensagens com o mesmo ID de grupo de mensagens, as filas FIFO poderão não retornar as mensagens que têm um ID de grupo de mensagens diferente, mas foram enviadas para a fila posteriormente até que você consuma com êxito as mensagens do backlog.

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](#) (p. 41).

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 diferentes [Filas virtuais](#) (p. 48) Com a mesma fila de host bloqueiem umas às outras, evite reutilizar o mesmo ID de grupo de mensagens com filas virtuais.

Uso do 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 do `ReceiveMessageChamadas`.

Durante uma interrupção de rede de longa duração que causa problemas de conectividade entre o SDK e o Amazon SQS, é uma melhor prática fornecer o ID de tentativa de solicitação de recebimento e repetir com o mesmo ID de tentativa de solicitação de recebimento, se a operação do SDK falhar.

Exemplos de SDK 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 da AWS. Para instalar e configurar o SDK, consulte [Conceitos básicos](#) no Guia do desenvolvedor do AWS SDK para 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 para 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 no lado do servidor \(SSE\) \(p. 60\)](#)
- [Configurar tags para uma fila \(p. 62\)](#)
- [Envio de atributos de mensagens \(p. 63\)](#)
- [Gerenciamento de mensagens grandes do Amazon SQS usando o Amazon S3 \(p. 65\)](#)

Usando criptografia no lado do servidor (SSE)

É possível usar o AWS SDK for Java para adicionar criptografia no lado do servidor (SSE) a uma fila do Amazon SQS. Cada fila usa uma chave mestra de cliente (CMK) do AWS Key Management Service (AWS KMS) para gerar as chaves de criptografia de dados. Este exemplo usa a CMK gerenciada pela AWS para o Amazon SQS. Para obter mais informações sobre como usar a SSE e a função do CMK, consulte [Criptografia em repouso \(p. 106\)](#).

Adicionando SSE a uma fila existente

Para ativar a criptografia no lado do servidor para uma fila existente, use o `SetQueueAttributes` para definir o método `KmsMasterKeyIdAttribute`.

O código de exemplo a seguir define a chave mestra como CMK gerenciada pela AWS para o Amazon SQS. O exemplo também define o [Período de reutilização de chaves mestra \(p. 107\)](#) para 140 segundos.

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

```
// 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 CMK 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);
```

Desativando o SSE para uma fila

Para desabilitar a criptografia no lado do servidor para uma fila existente, defina `aKmsMasterKeyIdAtributo` a uma string vazia usando o `SetQueueAttributes` Método do.

Important

`null` não é um valor válido para `KmsMasterKeyId`.

Criação de uma fila com SSE

Para ativar a SSE ao criar a fila, adicione `aKmsMasterKeyIdAtributo` para o `CreateQueue` Método de API.

O exemplo a seguir cria uma nova fila com SSE habilitado. A fila usa a CMK gerenciada pela AWS para o Amazon SQS. O exemplo também define o [Período de reutilização de chaves mestra \(p. 107\)](#) para 160 segundos.

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

```
// 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 CMK 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);
```

Recuperando atributos do SSE

Para obter informações sobre como recuperar atributos da fila, consulte [Exemplos](#) no Referência da API do Amazon Simple Queue Service.

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

Configurar tags para uma fila

Use tags de alocação de custos para ajudar a organizar e identificar suas filas do Amazon SQS. Os exemplos a seguir mostram como configurar tags usando o AWS SDK for Java. Para obter mais informações, consulte [Tag de alocação de custos do Amazon SQS](#) (p. 39).

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

Listar tags

Para listar as tags de uma fila, use a `tagListQueueTags` Método do.

```
// 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() ));
```

Adição ou atualização de tags

Para adicionar ou atualizar valores de tag para uma fila, use o `TagQueue` Método do.

```
// 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 do, use a `tagUntagQueueMétodo` do. O exemplo a seguir remove a propriedade `Accounting ID` Tag.

```
// 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 mensagens

Você pode incluir metadados estruturados (como carimbos de data e hora, dados geoespaciais, assinaturas e identificadores) com mensagens usando o `Atributos de mensagens` do. Para obter mais informações, consulte [Atributos de mensagem do Amazon SQS \(p. 35\)](#).

Antes de executar o código de exemplo, verifique se você definiu suas credenciais da AWS. Para obter mais informações, consulte [Configurar as credenciais e a região da AWS para desenvolvimento](#) no Guia do desenvolvedor do AWS SDK para Java 2.x.

Definindo atributos

Para definir um atributo para uma mensagem, adicione o código a seguir, que usa o `MessageAttributeValue` Tipo de dados. Para obter mais informações, consulte [Componentes de atributos de mensagem \(p. 35\)](#) e [Tipos de dados de atributos de mensagem \(p. 36\)](#).

O AWS SDK for Java calcula automaticamente as somas de verificação do corpo da mensagem e dos atributos da mensagem e compara-as com os dados retornados pelo Amazon SQS. Para obter mais informações, consulte o [Guia do desenvolvedor do AWS SDK para Java 2.x](#). [Cálculo do resumo de mensagens MD5 para atributos de mensagem \(p. 36\)](#) Para ver 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 `230.000000000000000001`.

```
final Map<String, MessageAttributeValue> messageAttributes = new HashMap<>();
messageAttributes.put("AccurateWeight", new MessageAttributeValue()
    .withDataType("Number")
    .withStringValue("230.000000000000000001"));
```

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 [ReceiveMessage](#) retorna método 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 ao `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 a uma fila FIFO, verifique se `sendMessage` o método executa. Depois, você fornecer o ID do grupo de mensagens.

Se você usar o `sendMessageBatch` método em vez de `sendMessage`, você deverá especificar os atributos de mensagens para cada mensagem no lote.

Gerenciamento de mensagens grandes do Amazon SQS usando o Amazon S3

Para gerenciar grandes mensagens do Amazon Simple Queue Service (Amazon SQS), você pode usar o Amazon Simple Storage Service (Amazon S3) e a Amazon SQS Extended Client Library para Java. Isso é especialmente útil para armazenar e consumir mensagens de até 2 GB. A menos que o aplicativo exija repetidamente a criação de filas deixando-as inativas ou armazenando grandes quantidades de dados nas filas, considere usar o Amazon S3 para armazenar seus dados.

Você pode usar a Amazon SQS Extended Client Library para Java 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
- Recuperar o objeto de mensagem de um bucket do S3
- Excluir o objeto de mensagem de um bucket do S3

Você pode usar a Amazon SQS Extended Client Library para Java para gerenciar mensagens do Amazon SQS usando o Amazon S3. Somente com o AWS SDK for Java. Você não pode fazer isso com a AWS CLI, o console do Amazon SQS, a API HTTP do Amazon SQS ou qualquer um dos outros SDKs da AWS.

OSDK para Java A Biblioteca de cliente estendida do Amazon SQS para Java exige o J2SE Development Kit 8.0 ou posterior.

Prerequisites

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

Antes de executar o código de exemplo, configure suas credenciais da AWS. Para obter mais informações, consulte [Configurar as credenciais e a região da AWS para desenvolvimento](#) no Guia do desenvolvedor do AWS SDK para Java.

Exemplo: Usando o Amazon S3 para gerenciar mensagens grandes do Amazon SQS

O exemplo a seguir cria um bucket do Amazon S3 com um nome aleatório e adiciona uma regra de ciclo de vida para excluir permanentemente objetos após 14 dias. Ele também cria uma fila chamada `MyQueue` e

envia uma mensagem aleatória 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.

```
/*
 * Copyright 2010-2021 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.sqs.javamessaging.AmazonSQSExtendedClient;
import com.amazonaws.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);
}
}
```

Como trabalhar com o JMS e o Amazon SQS

A Biblioteca de mensagens Java do Amazon SQS é uma interface JMS (Java Message Service) para Amazon SQS que permite aproveitar o Amazon SQS em aplicativos que já utilizam JMS. A interface permite que você use o Amazon SQS como o provedor JMS com o mínimo de alterações de 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 oferece suporte ao envio e recebimento de mensagens para uma fila (o modelo ponto a ponto 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 obter informações sobre os recursos da Biblioteca de mensagens Java do Amazon SQS que oferecem suporte à especificação JMS 1.1, consulte [Implementações JMS 1.1 com suporte \(p. 91\)](#) e [Perguntas frequentes do Amazon SQS](#).

Tópicos

- [Prerequisites \(p. 69\)](#)
- [Introdução à biblioteca de mensagens Java do Amazon SQS \(p. 70\)](#)
- [Usando o cliente do Amazon SQS Java Message Service \(JMS\) com outros clientes do Amazon SQS \(p. 76\)](#)
- [Java funcional de exemplo para uso do JMS com filas do Amazon SQS Standard \(p. 76\)](#)
- [Implementações JMS 1.1 com suporte \(p. 91\)](#)

Prerequisites

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.

[OSDK para Java](#)O Amazon SQS Extended Client Library for Java exige o J2SE Development Kit 8.0 ou posterior.

Para obter informações sobre como fazer download do SDK for Java, consulte [SDK para 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 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 a [Spring Framing](#).

Para exemplos de código que usam Maven, Spring Framework e Amazon SQS Java Messaging Library, consulte [Java funcional de exemplo para uso do JMS com filas do Amazon SQS Standard](#) (p. 76).

```
<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, o `CreateQueue` API ou o cliente do Amazon SQS do encapsulado incluído na Biblioteca de mensagens Java do Amazon SQS.

- Para obter informações sobre a criação de uma fila com o Amazon SQS usando o AWS Management Console ou o `CreateQueue` API, consulte [Criação de uma fila](#) (p. 12).
- Para obter informações sobre como usar a biblioteca de mensagens Java do Amazon SQS, consulte [Introdução à biblioteca de mensagens Java do Amazon SQS](#) (p. 70).

Introdução à 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 desta seção. As seções a seguir mostram como criar uma conexão e uma sessão JMS, e como enviar e receber uma mensagem.

O objeto de cliente do Amazon SQS incluído na biblioteca de mensagens Java do Amazon SQS verifica se existe uma fila do Amazon SQS. 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. No entanto, o `getWrappedAmazonSQSClient` método também fornece uma versão encapsulada do cliente Amazon SQS usada pela conexão atual. O wrapper transforma cada exceção de um cliente em um

`JMSEException`, permitindo que ele seja mais facilmente usado pelo código existente que espera ocorrências de `JMSEException`.

2. Você pode usar os objetos de cliente retornados `degetAmazonSQSClient` e `getWrappedAmazonSQSClient` para executar operações administrativas não incluídas na especificação 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.

Criação de uma fila do Amazon SQS

O objeto de cliente encapsulado verifica se existe uma fila do Amazon SQS.

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](#) (p. 78).

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 `.fifo` sufixo.

Para obter mais informações sobre o atributo `ContentBasedDeduplication`, consulte [Processamento exatamente uma vez](#) (p. 31).

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 `oAUTO_ACKNOWLEDGE` modo do.

```
// 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 a 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 a 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 obter mais informações, consulte [Principais termos \(p. 30\)](#).

```
// 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 do FIFO, como o ID do grupo de mensagens, o ID de eliminação de duplicação de mensagem e o número de sequência. Para obter mais informações, consulte [Principais termos \(p. 30\)](#).

```
// 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 [Java funcional de exemplo para uso do JMS com filas do Amazon SQS Standard \(p. 76\)](#).

Para exemplos completos de envio e recebimento de objetos, consulte [TextMessageSender.java \(p. 78\)](#) e [SyncMessageReceiver.java \(p. 80\)](#).

Recebimento de mensagens de forma assíncrona

No exemplo em [Introdução à biblioteca de mensagens Java do Amazon SQS \(p. 70\)](#), 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 [Introdução à biblioteca de mensagens Java do Amazon SQS \(p. 70\)](#). Para um exemplo completo de um consumidor assíncrono, consulte `AsyncMessageReceiver.java` em [Java funcional de exemplo para uso do JMS com filas do Amazon SQS Standard \(p. 76\)](#).

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 [Introdução à biblioteca de mensagens Java do Amazon SQS \(p. 70\)](#) O usa `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 [Introdução à biblioteca de mensagens Java do Amazon SQS \(p. 70\)](#). Para um exemplo completo de um consumidor síncrono com modo de reconhecimento do cliente, consulte `SyncMessageReceiverClientAcknowledge.java` em [Java funcional de exemplo para uso do JMS com filas do Amazon SQS Standard \(p. 76\)](#).

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 obter mais informações, consulte [Uso do modo de reconhecimento do cliente \(p. 74\)](#).

A Biblioteca de mensagens Java 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 \(p. 74\)](#). 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 cliente do Amazon SQS Java Message Service (JMS) com outros clientes do Amazon SQS

O uso do cliente do Amazon SQS Java Message Service (JMS) com o AWS SDK limita o tamanho da mensagem do Amazon SQS para 256 KB. No entanto, você pode criar um provedor JMS usando qualquer cliente do Amazon SQS. Por exemplo, você pode usar o JMS Client com o Amazon SQS Extended Client Library for Java para enviar uma mensagem do Amazon SQS que contenha uma referência à carga da mensagem (até 2 GB) no Amazon S3. Para obter mais informações, consulte [Gerenciamento de mensagens grandes do Amazon SQS usando o Amazon S3](#) (p. 65).

O seguinte exemplo de código Java cria o provedor JMS da Biblioteca do Extended Client:

```
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();
```

Java funcional de exemplo para uso do JMS com filas do Amazon SQS Standard

Veja a seguir exemplos de código que mostram como usar 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](#) (p. 71), [Envio de mensagens de forma síncrona](#) (p. 71), e [Recebimento de mensagens](#)

de forma síncrona (p. 72). 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 seguinte exemplo de código Java define o nome de fila padrão, a região e as credenciais a serem usadas com outros exemplos de Java.

```
/*
 * Copyright 2010-2021 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 2010-2021 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 JMSEException {
        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 (JMSEException 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 2010-2021 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 JMSEException {
        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 (JMSEException 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 2010-2021 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 JMSEException, 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() ) );

        ReceiverCallback callback = new ReceiverCallback();
        consumer.setMessageListener( callback );

        // No messages are processed until this is called
        connection.start();

        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 (JMSEException 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 2010-2021 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 JMSEException, 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 JMSEException
     */
    private static void sendMessage(MessageProducer producer, Session session, String
messageText) throws JMSEException {
        // 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 JMSEException
     */
    private static void receiveMessage(MessageConsumer consumer, boolean acknowledge)
throws JMSEException {
        // 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 2010-2021 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 JMSEException, 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 JMSEException
 */
private static void sendMessage(MessageProducer producer, Session session, String
messageText) throws JMSEException {
    // 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 JMSEException
 */
private static void receiveMessage(MessageConsumer consumer, boolean acknowledge)
throws JMSEException {
    // 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](#) (p. 88).

```
<!--
  Copyright 2010-2021 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 2010-2021 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 JMSEException {
        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 ( JMSEException 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 cria uma se não existir. Ele também inclui código de registro de exemplo.

```
/*
 * Copyright 2010-2021 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
    JMSEException {
        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 JMSEException {
            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() );
    }
}
}
```

Implementações JMS 1.1 com suporte

A biblioteca de mensagens Java do Amazon SQS oferece suporte ao seguinte [Implementações JMS 1.1](#). Para obter mais informações sobre os recursos e funcionalidades compatíveis da Biblioteca de mensagens Java do Amazon SQS, consulte o [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 explicitamente reconheça 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)

Depois de enviar mensagens, o Amazon SQS define os seguintes cabeçalhos e propriedades para cada mensagem:

- `JMSMessageID`
- `JMS_SQS_SequenceNumber`(somente para filas FIFO)

Para receber mensagens

Ao receber mensagens, o Amazon SQS define os seguintes cabeçalhos e propriedades para cada mensagem:

- `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 recursos e funcionalidades do Amazon SQS.

Tópicos

- [Criação de uma fila do Amazon SQS \(AWS CloudFormation\) \(p. 93\)](#)
- [Tutorial: Enviar uma mensagem para uma fila do Amazon SQS do Amazon Virtual Private Cloud \(p. 94\)](#)

Criação de uma fila do Amazon SQS (AWS CloudFormation)

Você pode usar o console do AWS CloudFormation, ou um modelo JSON (ou YAML) para criar uma fila do Amazon SQS. Para obter mais informações, consulte [Como trabalhar com modelos do AWS CloudFormation](#) e [a `AWS::SQS::Queue` no Guia do usuário do AWS CloudFormation](#).

Como usar o 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 o `FifoQueueContentBasedDeduplication` Propriedades. Para obter mais informações sobre a eliminação de duplicação baseada em conteúdo, consulte [Processamento exatamente uma vez \(p. 31\)](#).

Note

O nome de uma fila FIFO deve terminar com o `.fifo` Sufixo.

```
{
  "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](#), depois, escolha 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, selecione Next.
6. Na página Review (Revisar), selecione Create (Criar).

O AWS CloudFormation começa a criar o `MyQueue` e exibe o `CREATE_IN_PROGRESS` Status. Quando o processo é concluído, o AWS CloudFormation exibe o `CREATE_COMPLETE` Status.

Filter: Active ▾ By Stack Name		Showing 1 stack		
	Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/>	MyQueue	2017-02-20 11:39:47 UTC-0800	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: Enviar uma mensagem para uma fila do Amazon SQS do Amazon Virtual Private Cloud

Neste tutorial, você aprende 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 um VPC endpoint de interface, 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 obter mais informações, consulte [Amazon Virtual Private Cloud para o Amazon SQS \(p. 113\)](#).

Important

- Você pode usar o Amazon Virtual Private Cloud somente com endpoints HTTPS Amazon SQS.
- Ao configurar o Amazon SQS para enviar mensagens pela Amazon VPC, você precisa habilitar o DNS privado e especificar 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 key pair do Amazon EC2 \(p. 95\)](#)
- [Etapa 2: Criar recursos da AWS \(p. 95\)](#)

- [Etapa 3: Confirme que sua instância do EC2 não é acessível publicamente \(p. 96\)](#)
- [Etapa 4: Criar um Amazon VPC endpoint para o Amazon SQS \(p. 97\)](#)
- [Etapa 5: Envie uma mensagem para sua fila do Amazon SQS \(p. 97\)](#)

Etapa 1: Criar um key pair do Amazon EC2

Um par de chaves do O 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 recursos da AWS

Para configurar a infraestrutura necessária, você deve usar um AWS CloudFormation Modelo do, que é um esquema para criar uma pilha composta por recursos da AWS, 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
- Fila do Amazon SQS

1. Faça download do modelo do AWS CloudFormation chamado [SQS-VPCE-Tutorial-CloudFormation.yaml](#) do GitHub.
2. Faça login no [Console do AWS CloudFormation](#).
3. Selecione Criar Stack.
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 (Nome da chave), escolha SQS-VPCE-Tutorial-Key-Pair.
 - c. Escolha Next (Próximo).
6. Na página Options, selecione Next.
7. Na página Review (Revisar), na seção Capabilities (Recursos), selecione I acknowledge that AWS CloudFormation might create IAM resources with custom names. (Reconheço que o AWS CloudFormation pode criar recursos do IAM com nomes personalizados.) e, então, selecione Create (Criar).

O AWS CloudFormation começa a criar a pilha e exibe a CREATE_IN_PROGRESS Status. Quando o processo é concluído, o AWS CloudFormation exibe a CREATE_COMPLETE Status.

Etapa 3: Confirme que sua instância do EC2 não é acessível publicamente

Seu modelo do AWS CloudFormation executa uma instância do EC2 chamada SQS-VPCE-Tutorial-EC2-Instance em sua VPC. Esta instância do EC2 não permite tráfego de saída e não é capaz de enviar mensagens para o Amazon SQS. Para verificar isso, você precisa conectar-se à instância, tentar conectar-se a um endpoint público e, então, tentar enviar uma mensagem ao 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-EC2 Instance.
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 \(p. 95\)](#), conecte-se à instância usando o comando a seguir, por exemplo:

```
ssh -i SQS-VPCE-Tutorial-KeyPair.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 por seu modelo do AWS CloudFormation, por exemplo, VPCE-SQS-Tutorial-Stack-CFQueue-1ABCDEFH2IJK.
9. No Detalhes, copie a 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 VPC endpoint para o Amazon SQS, sua tentativa de envio será bem-sucedida.

Etapa 4: Criar um Amazon VPC endpoint para o Amazon SQS

Para conectar a VPC ao Amazon SQS, você precisa definir um VPC endpoint de interface. 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 dentro da rede da AWS sem precisar passar pela internet pública.

Note

A instância do EC2 ainda não tem acesso a outros serviços e endpoints da AWS na internet.

1. Faça login no [console da Amazon VPC](#).
2. No menu de navegação, selecione Endpoints.
3. Escolha Create Endpoint.
4. NoCriar endpoint, paraNome do serviço, escolha o nome do serviço para o Amazon SQS.

Note

Os nomes de serviços variam de acordo com a atual região da AWS. Por exemplo, se você estiver no Leste dos EUA (Ohio), o nome do serviço serácom.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 Create endpoint (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 oEndpoints.

A Amazon VPC começa a criar o endpoint e exibe opendenteStatus. Quando o processo estiver concluído, a Amazon VPC exibirá aavailableStatus.

Etapa 5: Envie uma mensagem para sua fila do Amazon SQS

Agora que a VPC inclui um endpoint para o Amazon SQS, você pode conectar-se à instância do EC2 e enviar mensagens para a fila.

1. Reconecte-se à instância do EC2, por exemplo:

```
ssh -i SQS-VPCE-Tutorial-KeyPair.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 por seu modelo do AWS CloudFormation (por exemplo, [VPCE-SQS-Tutorial-Stack-CFQueue-1ABCDEFGH2IJK](#)), consulte [Recebendo e excluindo mensagens \(console\)](#) (p. 22).

Para obter informações sobre como excluir seus recursos, consulte o seguinte:

- [Excluir um VPC endpoint](#)noGuia do usuário da Amazon VPC
- [Excluir uma fila do Amazon SQS](#) (p. 23)
- [Encerre sua instância](#)noGuia do usuário do Amazon EC2 para instâncias do Linux
- [Exclusão de sua VPC](#)noGuia do usuário da Amazon VPC
- [Excluir uma pilha no console do AWS CloudFormation](#)noGuia do usuário do AWS CloudFormation
- [Excluir o par de chaves](#)noGuia do usuário do Amazon EC2 para instâncias do Linux

Cotas do Amazon SQS

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

Tópicos

- [Cotas relacionadas a filas \(p. 99\)](#)
- [Cotas relacionadas a mensagens \(p. 100\)](#)
- [Cotas relacionadas a políticas \(p. 102\)](#)

Cotas relacionadas a filas

A tabela a seguir lista as cotas relacionadas a filas.

Quota	Descrição
Fila de atraso	O atraso padrão (mínimo) para uma fila é 0 segundos. 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 a maioria das filas padrão (dependendo do tráfego da fila e do acúmulo de mensagens), pode haver um máximo de aproximadamente 120.000 mensagens em processamento (recebidas de uma fila por um consumidor, mas ainda não excluídas da fila). Se você atingir essa cota ao usar o Sondagem curta (p. 40) , o Amazon SQS retorna o <code>overLimit</code> mensagem de erro. Se você usar sondagem longa (p. 40) , o Amazon SQS não retorna 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 um máximo de 20.000 mensagens em processamento (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.
Nome da fila	O nome da fila pode ter até 80 caracteres. Os seguintes caracteres são aceitos: caracteres alfanuméricos, hifens (-) e sublinhados (_).

Quota	Descrição
	Note Os nomes de fila diferenciam maiúsculas e minúsculas (por exemplo, <code>Test-queue</code> e <code>test-queue</code> são filas diferentes). O nome de uma fila FIFO deve terminar com o <code>.fifo</code> Sufixo. O sufixo conta para a cota de nome de fila de 80 caracteres. Para determinar se uma fila é FIFO (p. 28) , você pode verificar se o nome da fila termina com o sufixo.
Tag de fila	Não recomendamos adicionar mais de 50 tags a uma fila.
	A tag <code>Key</code> é necessária, mas a tag <code>Value</code> é opcional.
	A tag <code>Key</code> e a tag <code>Value</code> diferenciam maiúsculas de minúsculas.
	A tag <code>Key</code> e a tag <code>Value</code> podem incluir caracteres alfanuméricos Unicode em UTF-8 e espaços em branco. Os seguintes caracteres especiais são permitidos: <code>_ . : / = + - @</code>
	A tag <code>Key</code> ou <code>Value</code> não pode incluir o prefixo reservado <code>aws:</code> (não é possível excluir chaves ou valores de tag com esse prefixo).
	O comprimento máximo da tag <code>Key</code> é de 128 caracteres Unicode em UTF-8. A tag <code>Key</code> não pode estar vazia nem ser nula.
	O comprimento máximo da tag <code>Value</code> é de 256 caracteres Unicode em UTF-8. A tag <code>Value</code> pode estar vazia ou ser nula.
	As ações de marcação são limitadas a 5 TPS por conta da AWS. Se seu aplicativo exigir uma taxa de transferência mais alta, Envie uma solicitação .

Cotas relacionadas a mensagens

O alto throughput para as filas FIFO do Amazon SQS está na versão de teste e sujeito a alterações. Para obter mais informações, consulte [Antevisão: Alta taxa de transferência para filas FIFO \(p. 11\)](#).

A tabela a seguir lista as cotas relacionadas a mensagens.

Quota	Descrição
ID de mensagem em lote	O ID de uma mensagem em lote pode ter até 80 caracteres. Os seguintes caracteres são aceitos: caracteres alfanuméricos, hífens (<code>-</code>) e sublinhados (<code>_</code>).
Atributos de mensagem	Uma mensagem pode conter até 10 atributos de metadados.

Quota	Descrição
Lote de mensagens	Uma única solicitação em lote de mensagens pode incluir um máximo de 10 mensagens. Para mais informações, consulte Configurar o AmazonSQSBufferedAsyncClient (p. 168) na seção Ações de lote do Amazon SQS (p. 166) .
Conteúdo da mensagem	<p>Uma mensagem pode incluir apenas XML, JSON e texto não formatado. Os caracteres Unicode seguintes são permitidos: <code>#x9 #xA #xD #x20para#xD7FF #xE000para#xFFFD #x10000para#x10FFFF</code>.</p> <p>Os caracteres não incluídos nesta lista serão rejeitados. Para obter mais informações, consulte a Especificação W3C para caracteres.</p>
ID do grupo de mensagens	Consoma mensagens do backlog para evitar o acúmulo de um grande backlog de mensagens com o mesmo ID de grupo de mensagens (p. 59) .
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	<p>As filas padrão oferecem suporte a um número quase ilimitado de chamadas de API por segundo, por ação de API (SendMessage, ReceiveMessage ou DeleteMessage).</p> <p>Filas FIFO</p> <ul style="list-style-type: none"> Se você usa o agrupamento em lote (p. 166), as filas FIFO oferecem suporte a até 3.000 transações por segundo, por método de API (SendMessageBatch, ReceiveMessage ou DeleteMessageBatch). As 3000 transações 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). A visualização de alto throughput oferece até 3.000 transações por segundo (TPS) para mensagens em filas FIFO em regiões da AWS compatíveis. Para obter mais informações, consulte Antevisão: Alta taxa de transferência para filas FIFO (p. 11).
Temporizador de mensagem	O atraso padrão (mínimo) para uma mensagem é 0 segundos. O máximo é 15 minutos.
Tamanho da mensagem	<p>O tamanho mínimo da mensagem é de 1 byte (1 caractere). O comprimento máximo é de 262.144 bytes (256 KB).</p> <p>Para enviar mensagens maiores que 256 KB, você pode usar a Biblioteca de cliente estendida do Amazon SQS para Java. Essa biblioteca permite que você envie uma mensagem do Amazon SQS que contém uma referência à carga de uma mensagem no Amazon S3. O tamanho máximo de carga é 2 GB.</p>

Quota	Descrição
Tempo limite de visibilidade da mensagem	O tempo limite de visibilidade padrão para uma mensagem é de 30 segundos. O mínimo é de 0 segundos. O máximo é de 12 horas.
Informações de política	A cota máxima é 8.192 bytes, 20 declarações, 50 principais ou 10 condições. Para obter mais informações, consulte Cotas relacionadas a políticas (p. 102) .

Cotas relacionadas a políticas

A tabela a seguir lista as cotas relacionadas a políticas.

Nome	Máximo
Bytes	8,192
Condições	10
Principais	50
Declarações	20

Automação e solução de problemas de filas do Amazon SQS

Esta seção fornece informações sobre a automação e a solução de problemas de filas do Amazon SQS.

Tópicos

- [Automatização de notificações dos serviços da AWS para o Amazon SQS usando o Amazon EventBridge \(p. 103\)](#)
- [Solução de problemas do Amazon Simple Queue Service usando o AWS X-Ray \(p. 103\)](#)

Automatização de notificações dos serviços da AWS para o Amazon SQS usando o Amazon EventBridge

O Amazon EventBridge permite automatizar os serviços da AWS e responder aos eventos do sistema, como problemas de disponibilidade do aplicativo ou alterações de recursos. Os eventos dos serviços da AWS são entregues ao EventBridge quase em tempo real. Você pode 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.

O EventBridge permite que você defina uma variedade de destinos— como o Amazon SQS padrão e as filas FIFO — que recebem eventos no formato JSON. Para obter mais informações, consulte o [Guia do usuário do Amazon EventBridge](#).

Solução de problemas do Amazon Simple Queue Service usando o AWS X-Ray

O AWS X-Ray coleta dados sobre solicitações que seu aplicativo atende e permite visualizar e filtrar dados para identificar possíveis problemas e oportunidades de otimização. Para qualquer solicitação rastreada para a aplicação, é possível ver informações detalhadas sobre a solicitação, a resposta e as chamadas que a aplicação faz para recursos, microsserviços, bancos de dados e APIs da web HTTP.

Para enviar cabeçalhos de rastreamento do AWS X-Ray por meio do Amazon SQS, é possível executar uma das seguintes ações:

- Usar `oX-Amzn-Trace-Id` [Cabeço de rastreamento](#).
- Usar `oAWSXRayTraceHeader` [Atributo do sistema do \(p. 37\)](#).

Para coletar dados sobre erros e latência, é necessário instrumentar o cliente [AmazonSQS](#) usando o [SDK do AWS X-Ray](#).

É possível usar o AWS X-Ray para visualizar o mapa de conexões entre o Amazon SQS e outros serviços usados pelo aplicativo. 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-Ray](#) no 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.

Tópicos

- [Proteção de dados \(p. 105\)](#)
- [Identity and Access Management no Amazon SQS \(p. 114\)](#)
- [Registrar em log e monitorar no Amazon SQS \(p. 143\)](#)
- [Validação de conformidade do Amazon SQS \(p. 156\)](#)
- [Resiliência no Amazon SQS \(p. 156\)](#)
- [Segurança da infraestrutura no Amazon S3 \(p. 157\)](#)
- [Melhores práticas de segurança do Amazon SQS \(p. 157\)](#)

Proteção de dados

A [AWS Modelo de responsabilidade compartilhada](#) se aplica à proteção de dados no Amazon Simple Queue Service. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos serviços da AWS que você usa. Para obter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#). Para obter informações sobre a proteção de dados na Europa, consulte o [Modelo de responsabilidade compartilhada da AWS e GDPR](#) Publicação no blog do Segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da conta da AWS e configure contas de usuário individuais com o 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 da AWS. Recomendamos TLS 1.2 ou posterior.
- Configure o registro em log de atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão nos serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados pessoais armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para obter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como números de conta dos seus clientes, em campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalhar com o Amazon SQS ou outros serviços da AWS usando o console, a API, a CLI da AWS ou os AWS SDKs. Todos os dados que você insere no Amazon SQS ou em outros serviços podem ser separados para inclusão em logs de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar a solicitação a esse servidor.

As seções a seguir fornecem mais informações sobre a proteção de dados no Amazon SQS.

Tópicos

- [Criptografia de dados \(p. 106\)](#)
- [Privacidade do tráfego entre redes \(p. 112\)](#)

Criptografia de dados

Proteção de dados protege 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. Para proteger dados em repouso, solicite que o Amazon SQS criptografe as mensagens antes de salvá-las nos discos dos datacenters e as descriptografe quando as mensagens forem recebidas.

Tópicos

- [Criptografia em repouso \(p. 106\)](#)
- [Gerenciamento de chaves \(p. 108\)](#)

Criptografia em repouso

A criptografia no lado do servidor (SSE) permite que você transmita dados confidenciais em filas criptografadas. A SSE protege o conteúdo de mensagens em filas do usando chaves gerenciadas no AWS Key Management Service (AWS KMS). Para obter informações sobre como gerenciar o SSE usando o Console de Gerenciamento da AWS, consulte [Configurar a criptografia no lado do servidor \(SSE\) para uma fila \(console\)](#) (p. 16).

Para obter informações sobre como gerenciar o SSE usando o AWS SDK for Java (e `CreateQueue`, `SetQueueAttributes`, e `GetQueueAttributes`), consulte os seguintes exemplos:

- [Usando criptografia no lado do servidor \(SSE\) \(p. 60\)](#)
- [Configurar permissões do KMS para serviços da AWS \(p. 109\)](#)

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 usuário autorizado.

Important

Todas as solicitações para filas com SSE ativada devem usar HTTPS e [Signature versão 4](#). Uma [fila criptografada \(p. 106\)](#) que usa a chave padrão (CMK gerenciado pela AWS para Amazon SQS) não pode invocar uma função do Lambda em uma conta da AWS diferente. Alguns recursos dos serviços da AWS que podem enviar notificações para o Amazon SQS usando o AWS Security Token Service [AssumeRole](#) são compatíveis com SSE, mas funcionam somente com filas padrão:

- [Ganchos de ciclo de vida de auto escala](#)
- [Dead-letter do AWS Lambda](#)

Para obter informações sobre a compatibilidade de outros serviços com filas criptografadas do, consulte [Configurar permissões do KMS para serviços da AWS \(p. 109\)](#) e sua documentação de serviço.

O AWS KMS integra hardware e software seguros e altamente disponíveis para oferecer um sistema de gerenciamento de chaves escalonado para a nuvem. Quando você usa o Amazon SQS com o AWS KMS, o [Chaves de dados \(p. 107\)](#) A criptografar os dados da mensagem também são criptografados e armazenados com os dados que eles protegem.

Veja a seguir os benefícios do uso do AWS KMS:

- Você mesmo pode criar e gerenciar as [Customer master keys \(CMKs – Chaves mestras de cliente\)](#) (p. 107).
- Você também pode usar a CMK gerenciada pela AWS para o Amazon SQS, que é exclusiva para cada conta e região.
- Os padrões de segurança da AWS KMS podem ajudá-lo a atender a 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](#) (p. 107)
- [Principais termos](#) (p. 107)

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. 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](#) (p. 41) não afeta sua criptografia:

- Quando o Amazon SQS transfere uma mensagem de uma fila de origem criptografada para uma deadletter queue não criptografada, a mensagem permanece criptografada.
- Quando o Amazon SQS transfere uma mensagem de uma fila de origem não criptografada para uma deadletter queue 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 o [Referência de API do Amazon Simple Queue Service](#).

Chave de dados

A chave de criptografia de dados (DEK) responsável por criptografar o conteúdo das 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 tempo, em segundos, durante o qual o Amazon SQS pode reutilizar uma chave de dados para criptografar ou descriptografar mensagens antes de chamar o AWS KMS novamente. Um número inteiro que representa segundos, entre 60 segundos (1 minuto) e 86.400 segundos (24 horas). O padrão é 300 (5 minutos). Para obter mais informações, consulte [Entender o período de reutilização de chaves de dados](#) (p. 111).

Note

No evento improvável de não poder acessar o AWS KMS, o Amazon SQS continua a usar a chave de dados armazenada em cache até que uma conexão seja restabelecida.

ID da chave mestra do cliente

O alias, o ARN do alias, o ID de chave ou o ARN da chave de uma chave mestra do cliente (CMK) gerenciada pela AWS ou uma CMK personalizada — em sua conta ou em outra conta. Embora o alias do CMK gerenciada pela AWS para o Amazon SQS seja sempre `alias/aws/sqs`, o alias de uma CMK personalizada pode, por exemplo, ser `alias/MyAlias`. Você pode usar essas CMKs para proteger as mensagens nas filas do Amazon SQS.

Note

Lembre-se do seguinte:

- Se você não especificar uma CMK personalizada, o Amazon SQS usará a CMK gerenciada pela AWS para o Amazon SQS.
- Na primeira vez que você usa o AWS Management Console para especificar o CMK gerenciado pela AWS para o Amazon SQS para uma fila, o AWS KMS cria o CMK gerenciado pela AWS para o Amazon SQS.
- Alternativamente, na primeira vez que você usar `SendMessage` ou `SendMessageBatch` em uma fila com SSE ativada, o AWS KMS cria a CMK gerenciada pela AWS para o Amazon SQS.

Você pode criar CMKs, definir as políticas que controlam como as CMKs podem ser usadas e auditar o uso da CMK utilizando o `Customer managed keys` do console do AWS KMS ou da `CreateKey` Ação do AWS KMS. Para obter mais informações, consulte [Chaves mestras de cliente \(CMKs\)](#) e [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service. Para obter mais exemplos de identificadores de CMK, consulte `KeyId` no Referência de API do AWS Key Management Service. Para obter informações sobre como encontrar identificadores de CMK, consulte [Encontrar o ID de chave e o ARN](#) no Guia do desenvolvedor do AWS Key Management Service.

Important

Há custos adicionais pelo uso de AWS KMS. Para obter mais informações, consulte [Estimativa de custos do AWS KMS](#) (p. 111) 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 CMK para criptografar a chave de dados e, depois, a chave de dados criptografada é armazenada com a mensagem criptografada. Essa prática de uso de uma chave mestra 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

O Amazon SQS integra-se ao AWS Key Management Service para gerenciarchaves mestras do cliente(CMKs) para criptografia do lado do servidor (SSE). Consulte [Criptografia em repouso](#) (p. 106) para

obter informações sobre SSE e definições de gerenciamento de chaves. O Amazon SQS usa CMKs para validar e proteger as chaves de dados que criptografam e descriptografam as mensagens. As seções a seguir fornecem mais informações sobre como trabalhar com CMKs e chaves de dados no serviço Amazon SQS.

Tópicos

- [Configurar permissões do AWS KMS \(p. 109\)](#)
- [Entender o período de reutilização de chaves de dados \(p. 111\)](#)
- [Estimativa de custos do AWS KMS \(p. 111\)](#)
- [AWS KMS \(p. 112\)](#)

Configurar permissões do AWS KMS

Cada CMK deve ter uma política de chave. Observe que não é possível modificar a política de chaves de uma CMK gerenciada pela AWS para o Amazon SQS. A política desse CMK inclui permissões de uso das filas criptografadas para todos os principais na conta (que estão autorizados a usar o Amazon SQS).

Para uma CMK 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 da CMK. Para obter mais informações sobre permissões do AWS KMS, consulte [Recursos e operações do AWS KMS](#) ou [Referência de permissões de API do AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

Como alternativa, você pode especificar as permissões necessárias em uma política do IAM atribuída às entidades principais que produzem e consomem mensagens criptografadas. Para obter mais informações, consulte [Uso de políticas do IAM com o 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, o AWS KMS requer a indicação explícita do ARN de CMKs em regiões específicas no `Resource` de uma política do IAM.

Configurar permissões do KMS para serviços da AWS

Vários serviços da AWS servem como origens de evento que podem enviar eventos para filas do Amazon SQS. Para permitir que essas origens de evento funcionem com filas criptografadas, é necessário criar um CMK gerenciado pelo cliente e adicionar permissões na política de chave para que o serviço use os métodos necessários de API do AWS KMS. Execute as etapas a seguir para configurar as permissões.

1. Crie uma CMK gerenciada pelo cliente. Para obter mais informações, consulte [Criar chaves](#) no Guia do desenvolvedor do AWS Key Management Service.
2. Para permitir que a origem do evento de serviço da AWS use `kms:GenerateDataKey` e `kms:Decrypt` Métodos de API, adicione a declaração a seguir à política de chave da CMK.

```
{
  "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
Amazon CloudWatch Events	events.amazonaws.com
Notificações de eventos do Amazon S3	s3.amazonaws.com
Assinaturas de tópicos do Amazon SNS	sns.amazonaws.com

3. [configurar uma fila SSE existente \(p. 16\)](#) usando o ARN do seu CMK.
4. Forneça o ARN da fila criptografada para a fonte do evento.

Configurar permissões do KMS para produtores

Quando o [período de reutilização da chave de dados \(p. 111\)](#) 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 mestra do cliente (CMK).

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 permissões do KMS 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 mestra do cliente (CMK) que é usada para criptografar as mensagens na fila específica. Se a fila agir como uma [dead letter queue \(p. 41\)](#), o consumidor também deverá ter a permissão `kms:Decrypt` para qualquer CMK 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"
  }]
}
```

Entender o período de reutilização de chaves de dados

O [Período de reutilização de chaves de dados \(p. 107\)](#) define a duração máxima do Amazon SQS para reutilizar a mesma chave de dados. 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 menor de reutilização fornece melhor segurança, mas resulta em mais chamadas para o AWS KMS, o que pode ser cobrado 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 terminar, a próxima chamada para `SendMessageBatch` normalmente aciona uma chamada para o `GenerateDataKey` para obter uma nova chave de dados. Além disso, as próximas chamadas para `SendMessage` e `ReceiveMessage` cada uma acionará uma chamada para o `Decrypt` para verificar a integridade da chave de dados antes de usá-la.
- **Principais** (contas da AWS ou usuários do IAM) não compartilham chaves de dados (mensagens enviadas por entidades principais exclusivas sempre obtêm chaves de dados exclusivas). Portanto, o volume de chamadas para o AWS KMS é um múltiplo do número de entidades principais exclusivas em uso durante o período de reutilização de chaves de dados:

Estimativa de custos do AWS KMS

Para prever custos e entender melhor sua fatura da AWS, é recomendável saber com que frequência o Amazon SQS usa sua chave mestra do cliente (CMK).

Note

Embora a fórmula a seguir possa dar a você uma excelente 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 \(p. 107\)](#) (em segundos).

P é o número de [principais](#) que enviam para a fila do Amazon SQS.

Cé o número de entidades principais 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 obter mais informações, consulte [Entender o período de reutilização de chaves de dados \(p. 111\)](#).

Se o produtor e o consumidor tiverem usuários diferentes do IAM, o custo aumentará.

Estes são cálculos de exemplo. Para obter informações exatas sobre definição de preço, consulte a [Definição de preço do AWS Key Management Service](#).

Exemplo 1: Calcular o número de chamadas à API do AWS KMS para duas entidades principais e uma 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: Calcular o número de chamadas à API do AWS KMS 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

Ao trabalhar com o Amazon SQS e o AWS KMS, você pode encontrar erros. As referências a seguir descrevem os erros e possíveis soluções de problemas.

- [Erros comuns do AWS KMS](#)
- [AWS KMS Decrypt errors](#)
- [AWS KMS GenerateDataKey](#)

Privacidade do tráfego entre redes

Um endpoint da Amazon Virtual Private Cloud (Amazon VPC) para Amazon SQS é uma entidade lógica em 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

- [Amazon Virtual Private Cloud para o Amazon SQS \(p. 113\)](#)
- [Criar uma política de VPC endpoint para o Amazon SQS \(p. 113\)](#)

Amazon Virtual Private Cloud para o Amazon SQS

Se você usa a Amazon VPC para hospedar seus recursos da AWS, pode estabelecer uma conexão entre a 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 executar recursos da AWS 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 mais informações sobre VPCs, consulte o [Guia do usuário da Amazon VPC](#).

Para conectar sua VPC ao Amazon SQS, você precisa primeiro definir uma VPC endpoint de interface, o que permite conectar sua VPC a outros serviços da AWS. O endpoint fornece conectividade confiável e escalável com o Amazon SQS sem a necessidade de um gateway da Internet, da instância de conversão de endereço de rede (NAT) ou de uma conexão VPN. Para obter mais informações, consulte [Tutorial: Enviar uma mensagem para uma fila do Amazon SQS do Amazon Virtual Private Cloud \(p. 94\)](#) e [Exemplo 5: Negar acesso se não vier de um VPC endpoint \(p. 139\)](#) neste guia e [VPC endpoints de interface \(AWS PrivateLink\)](#) no Guia do usuário da Amazon VPC.

Important

- Você pode usar o Amazon Virtual Private Cloud somente com endpoints HTTPS Amazon SQS.
- Ao configurar o Amazon SQS para enviar mensagens do Amazon VPC, é preciso ativar o DNS privado e especificar 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 VPC endpoint para o Amazon SQS

Você pode criar uma política para endpoints da Amazon VPC para o Amazon SQS na qual especifica-se o seguinte:

- O 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 [Controle do acesso a serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC

O exemplo de política de VPC endpoint policy especifica que o usuário do IAM `MyUser` tem permissão para enviar mensagens para a fila do Amazon SQS `MyQueue`.

```
{
  "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 do IAM e regras que tentam usar esse VPC endpoint.
- `MyUser` enviar mensagens para outra fila do Amazon SQS.

Note

O usuário do IAM ainda pode usar outras ações de API do Amazon SQS fora do VPC. Para obter mais informações, consulte [Exemplo 5: Negar acesso se não vier de um VPC endpoint](#) (p. 139).

Identity and Access Management no Amazon SQS

O acesso ao Amazon SQS exige credenciais que a AWS possa usar para autenticar suas solicitações. Essas credenciais devem ter permissões para acessar os recursos da AWS, como filas do Amazon SQS e mensagens do. As seguintes seções fornecem detalhes sobre como você pode usar o [AWS Identity and Access Management \(IAM\)](#) e o Amazon SQS para ajudar a proteger seus recursos controlando o acesso a eles.

Tópicos

- [Authentication](#) (p. 114)
- [Controle de acesso](#) (p. 115)
- [Visão geral do gerenciamento de acesso no Amazon SQS](#) (p. 116)
- [Uso de políticas baseadas em identidade com o Amazon SQS](#) (p. 121)
- [Uso de políticas personalizadas com a linguagem de política de acesso do Amazon SQS](#) (p. 129)
- [Uso de credenciais de segurança temporárias com o Amazon SQS](#) (p. 139)
- [Permissões de API do Amazon SQS: Referência de ações e recursos](#) (p. 141)

Authentication

Você pode acessar a AWS como alguns dos seguintes tipos de identidades:

- **Usuário raiz da conta da AWS:** ao criar uma conta AWS, você começa com uma única identidade de login que tem acesso completo a todos os serviços e recursos da AWS na conta. Essa identidade é denominada usuário raiz da conta da AWS e é acessada pelo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável que você não use o usuário raiz nas tarefas diárias, nem mesmo nas administrativas. Em vez disso, siga as [práticas recomendadas para o uso do usuário raiz somente a fim de criar seu primeiro usuário do IAM](#). Depois, armazene as credenciais do usuário raiz com segurança e use-as para executar somente algumas tarefas de gerenciamento de contas e de serviços.
- **Usuário do IAM**— Um [Usuário do IAM](#) é uma identidade na sua conta da AWS com permissões personalizadas específicas (por exemplo, permissões para criar uma fila no Amazon SQS). É possível usar um nome de usuário e uma senha do IAM para fazer login em páginas da Web seguras da AWS,

como o [Console de Gerenciamento da AWS](#), os [Fóruns de discussão da AWS](#) ou o [AWS Support Center](#).

Além de um nome e senha de usuário, você também pode gerar [chaves de acesso](#) para cada usuário. Você pode usar essas chaves ao acessar serviços da AWS de forma programática, seja por meio de um [dos vários SDKs](#) ou usando a [Interface da linha de comando \(CLI\) da AWS](#). As ferramentas de SDK e de CLI usam as chaves de acesso para o cadastramento criptográfico da sua solicitação. Se você não utilizar ferramentas da AWS, assine a solicitação você mesmo. O Amazon SQS oferece suporte a [Signature versão 4](#), um protocolo para autenticar solicitações de API de entrada. Para obter mais informações sobre solicitações de autenticação, consulte [Processo de assinatura do Signature versão 4](#) na Referência geral da AWS.

- **Função do IAM:** uma [função do IAM](#) é uma identidade do IAM que você pode criar em sua conta com permissões específicas. Uma função do IAM é semelhante a um usuário do IAM, pois é uma identidade da AWS com políticas de permissão que determinam o que a identidade pode e não pode fazer na AWS. No entanto, em vez de ser exclusivamente associada a uma pessoa, uma função destina-se a ser assumida por qualquer pessoa que precisar dela. Além disso, uma função não tem credenciais de longo prazo padrão, como uma senha ou chaves de acesso, associadas a ela. Em vez disso, quando você assumir uma função, ela fornecerá credenciais de segurança temporárias para sua sessão de função. As funções do IAM com credenciais temporárias são úteis nas seguintes situações:
- **Acesso de usuário federado:** em vez de criar um usuário do IAM, é possível usar identidades existentes do AWS Directory Service, o diretório de usuários da sua empresa ou um provedor de identidades da Web. Estes são conhecidos como usuários federados. A AWS atribui uma função a um usuário federado quando o acesso é solicitado por meio de um [provedor de identidades](#). Para obter mais informações sobre usuários federados, consulte [Usuários e funções federados](#) no Guia do usuário do IAM.
- **Acesso ao serviço da AWS:** uma função de serviço é uma [função do IAM](#) que um serviço assume para realizar ações em seu nome. As funções de serviço fornecem acesso apenas dentro de sua conta e não podem ser usadas para conceder acesso a serviços em outras contas. Um administrador do IAM pode criar, modificar e excluir uma função de serviço do IAM. Para obter mais informações, consulte [Criar uma função para delegar permissões a um serviço da AWS](#) no Guia do usuário do IAM.
- **Aplicações em execução no Amazon EC2:** é possível usar uma função do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 que fazem solicitações de API da AWS ou de CLI da AWS. É preferível fazer isso do que armazenar chaves de acesso na instância do EC2. Para atribuir uma função da AWS a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, crie um perfil de instância para ser anexado à instância. Um perfil de instância contém a função e permite que programas que estão em execução na instância do EC2 obtenham credenciais temporárias. Para obter mais informações, consulte [Usar uma função do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2](#) no Guia do usuário do IAM.

Controle de acesso

O Amazon SQS tem seu próprio sistema de permissões com base em recursos que usa políticas escritas na mesma linguagem usada para políticas do AWS Identity and Access Management (IAM). Isso significa que você pode alcançar objetivos semelhantes com as políticas do Amazon SQS e do IAM.

Note

É importante entender que todas as contas da AWS podem delegar suas permissões para usuários em suas contas. O acesso entre contas permite compartilhar o acesso aos seus recursos da AWS sem a necessidade de gerenciar usuários adicionais. Para obter mais informações sobre como usar o acesso entre contas, consulte [Ativar acesso entre contas](#) no Guia do usuário do IAM. As permissões entre contas não se aplicam às seguintes ações:

- [AddPermission](#)
- [CreateQueue](#)
- [DeleteQueue](#)
- [ListQueues](#)
- [ListQueueTags](#)
- [RemovePermission](#)
- [SetQueueAttributes](#)
- [TagQueue](#)
- [UntagQueue](#)

No momento, o Amazon SQS oferece suporte a um subconjunto limitado de [chaves de condição disponíveis no IAM](#). Para obter mais informações, consulte [Permissões de API do Amazon SQS: Referência de ações e recursos \(p. 141\)](#).

Visão geral do gerenciamento de acesso no Amazon SQS

Cada recurso da AWS é de propriedade de uma conta da AWS, e as permissões para criar ou acessar um recurso são regidas por políticas de permissões. Um administrador de conta pode anexar políticas de permissões a identidades do IAM (usuários, grupos e funções), e alguns serviços (como o Amazon SQS) também dão suporte à anexação de políticas de permissões a recursos.

Note

O administrador da conta (ou usuário administrador) é um usuário com privilégios administrativos. Para obter mais informações, consulte as [melhores práticas do 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 \(p. 116\)](#)
- [Entender a propriedade de recursos \(p. 117\)](#)
- [Gerenciar o acesso aos recursos \(p. 117\)](#)
- [Especificar elementos de política: Ações, efeitos, recursos e entidades principais \(p. 120\)](#)
- [Especificar condições em uma política \(p. 121\)](#)

Recursos e operações do Amazon Simple Queue Service

No Amazon SQS, o único recurso é o 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	<code>arn:aws:sqs:<i>region</i>:<i>account_id</i>:<i>queue_name</i></code>

Veja a seguir exemplos do formato do ARN para filas:

- Um ARN para uma fila denominada `my_queue` Na região Leste dos EUA (Ohio), pertencente à Conta da AWS 123456789012:

```
arn:aws:sqs:us-east-2:123456789012:my_queue
```

- Um ARN para uma fila denominada `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 [IAM ARNs](#) no Guia do usuário do IAM.

O Amazon SQS fornece um conjunto de ações de que funcionam com o recurso da fila. Para obter mais informações, consulte [Permissões de API do Amazon SQS: Referência de ações e recursos \(p. 141\)](#).

Entender a propriedade de recursos

A conta da AWS é proprietária dos recursos criados na conta, independentemente de quem os criou. Mais especificamente, o proprietário do recurso é a conta da AWS da entidade principal (ou seja, a conta raiz, um usuário do IAM; ou uma função do IAM) que autentica a solicitação de criação de recursos. Os exemplos a seguir ilustram como isso funciona:

- Se você usar as credenciais da conta raiz de sua conta da AWS para criar uma fila do Amazon SQS, sua conta da AWS será a proprietária do recurso (no Amazon SQS, o recurso é a fila do Amazon SQS).
- Se você criar um usuário do IAM na sua conta da AWS e conceder permissões para esse usuário criar uma fila do, ele poderá criar a fila. No entanto, a conta da AWS (à qual o usuário pertence) é a proprietária do recurso da fila.
- Se você criar uma função do IAM na sua conta da AWS com permissões para criar uma fila do Amazon SQS, qualquer pessoa que puder assumir a função poderá criar uma fila. Sua conta da AWS (à qual a função pertence) é a proprietária do recurso da fila.

Gerenciar o acesso aos recursos

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 obter a documentação completa do IAM, consulte [O que é](#)

o [IAM](#)?noGuia do usuário do IAM. Para obter mais informações sobre a sintaxe as descrições da política do IAM[Referência de política do AWS IAM](#)noGuia do usuário do IAM.

As políticas anexadas a uma identidade do IAM são conhecidas comoBaseado em identidadeAs políticas do IAM e as políticas anexadas a um recurso são chamadas deBaseado em recursos doPolíticas do.

Políticas baseadas em identidade (políticas do IAM e políticas do Amazon SQS)

Há duas maneiras de conceder aos usuários permissões às suas filas do Amazon SQS: usando o sistema 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 permissão a um usuário ou grupo na sua conta— Para conceder permissões de usuário 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 permissão a um usuário em outra conta da AWS— Para conceder permissões de usuário 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](#)
- [CreateQueue](#)
- [DeleteQueue](#)
- [ListQueues](#)
- [ListQueueTags](#)
- [RemovePermission](#)
- [SetQueueAttributes](#)
- [TagQueue](#)
- [UntagQueue](#)
- Anexar uma política de permissão 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 administrador da conta A da AWS pode criar uma função para conceder permissões entre contas à conta da AWS B (ou um serviço da AWS) da seguinte forma:
 - O administrador da conta A da cria uma função do IAM e anexa uma política de permissões à função, que concede permissões para os recursos na conta A.
 - 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 deseja conceder a permissão para assumir a função a um serviço da AWS, a entidade principal da política de confiança também pode ser a entidade principal de um serviço da AWS.

Para obter mais informações sobre o uso do IAM para delegar permissões, consulte[Gerenciamento de acesso](#)noGuia do usuário do IAM.

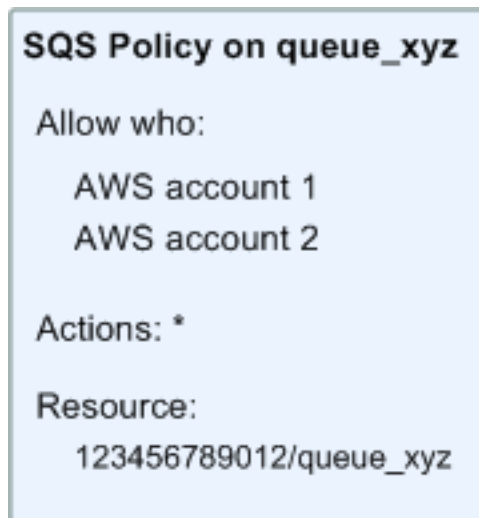
Embora o Amazon SQS funcione com políticas do IAM, ele tem sua própria infraestrutura de políticas. Você pode usar uma política do Amazon SQS com uma fila para especificar quais contas da AWS têm acesso à fila. Você pode especificar o tipo de acesso e condições (por exemplo, uma condição que conceda 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 escreve uma política do Amazon SQS e especifica *Para "permitir todas as ações do Amazon SQS", isso indica que um usuário pode executar todas as ações nesse subconjunto.

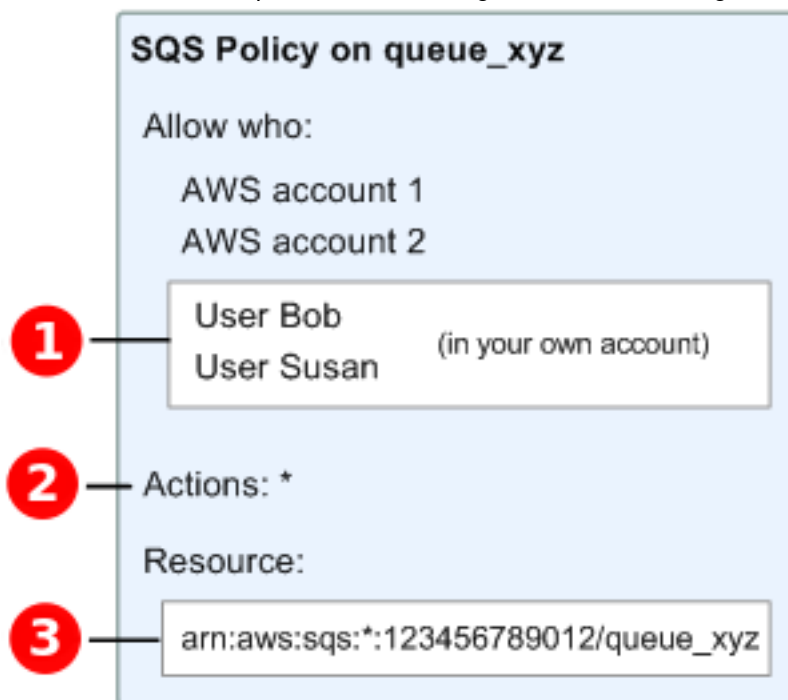
O diagrama a seguir ilustra o conceito de uma dessas políticas básicas do Amazon SQS que abrangem o subconjunto de ações. A política é para `queue_xyz` e fornece às contas 1 da AWS e à conta 2 da AWS permissões para usar qualquer uma das ações permitidas com a fila especificada.

Note

O recurso na política é especificado como `123456789012/queue_xyz`, onde `123456789012` é o ID da conta da AWS da que possui a fila.



Com a introdução do IAM e os conceitos de usuários e nomes de recurso da Amazon (ARN), algumas coisas mudaram sobre políticas SQS. O diagrama e a tabela a seguir descrevem as alterações.



❶ Para obter informações sobre como conceder permissões a usuários em diferentes contas, consulte [Tutorial: Delegar acesso em contas da AWS usando funções do IAM](#) no Guia do usuário do IAM.

❷ O subconjunto de ações incluídas em * foi expandido. Para obter uma lista de ações permitidas, consulte [Permissões de API do Amazon SQS: Referência de ações e recursos \(p. 141\)](#).

❸ Você pode especificar o recurso usando o nome de recurso da Amazon (ARN), a forma padrão de especificar recursos nas políticas do IAM. Para obter mais informações sobre o formato do ARN para filas do Amazon SQS, consulte [Recursos e operações do Amazon Simple Queue Service \(p. 116\)](#).

Por exemplo, de acordo com a política do Amazon SQS no diagrama anterior, qualquer pessoa que possua credenciais de segurança para a conta 1 da AWS ou a conta 2 da pode acessar `queue_xyz`. Além disso, os usuários Bob e Susan em sua própria conta da AWS (com o ID123456789012) pode acessar a fila.

Antes da introdução do IAM, o Amazon SQS concedia automaticamente ao criador de uma fila o controle total sobre a fila (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 um exemplo de política 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 [Uso de políticas baseadas em identidade com o Amazon SQS \(p. 121\)](#), e [identidades \(usuários, grupos e funções\)](#) no Guia do usuário do IAM.

Especificar elementos de política: Ações, efeitos, recursos e entidades principais

Para cada [Amazon Simple Queue Service \(p. 116\)](#), o serviço define um conjunto de [Ações do](#). Para conceder permissões a essas ações do, o Amazon SQS define um conjunto de ações que podem ser especificados 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 ações de recurso que deseja permitir ou negar. Por exemplo, as receitas `sqs:CreateQueue` permite que o usuário execute o Amazon Simple Queue Service `CreateQueue` Ação .
- **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 (aplica-se somente a políticas baseadas em recursos).

Para saber mais sobre a sintaxe e as descrições de políticas do Amazon SQS, consulte [Referência de política do AWS IAM](#) no Guia do usuário do IAM.

Para ver uma tabela com todas as ações do Amazon Simple Queue Service e os recursos aos quais se aplicam, consulte [Permissões de API do Amazon SQS: Referência de ações e recursos \(p. 141\)](#).

Especificar condições em uma política

Ao conceder permissões, você pode usar a linguagem da política de acesso do Amazon SQS para especificar as condições para quando uma política deve entrar em vigor. Por exemplo, convém que uma política só seja aplicada após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte [Condição](#) no Guia do usuário do IAM.

Para expressar condições, você usa chaves de condição predefinidas. Não há chaves de condição específicas do Amazon SQS. No entanto, existem chaves de condição em toda a AWS que você pode usar com o Amazon SQS. No momento, o Amazon SQS suporta apenas um conjunto limitado de chaves de condição disponíveis no IAM. Consulte [the section called “Referência de permissões da API” \(p. 141\)](#).

Uso de políticas baseadas em identidade 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 obter mais informações, consulte [Visão geral do gerenciamento de acesso no Amazon SQS \(p. 116\)](#).

Com exceção de `ListQueues`, todas as ações do Amazon SQS dão suporte a permissões no nível do recurso. Para obter mais informações, consulte [Permissões de API do Amazon SQS: Referência de ações e recursos \(p. 141\)](#).

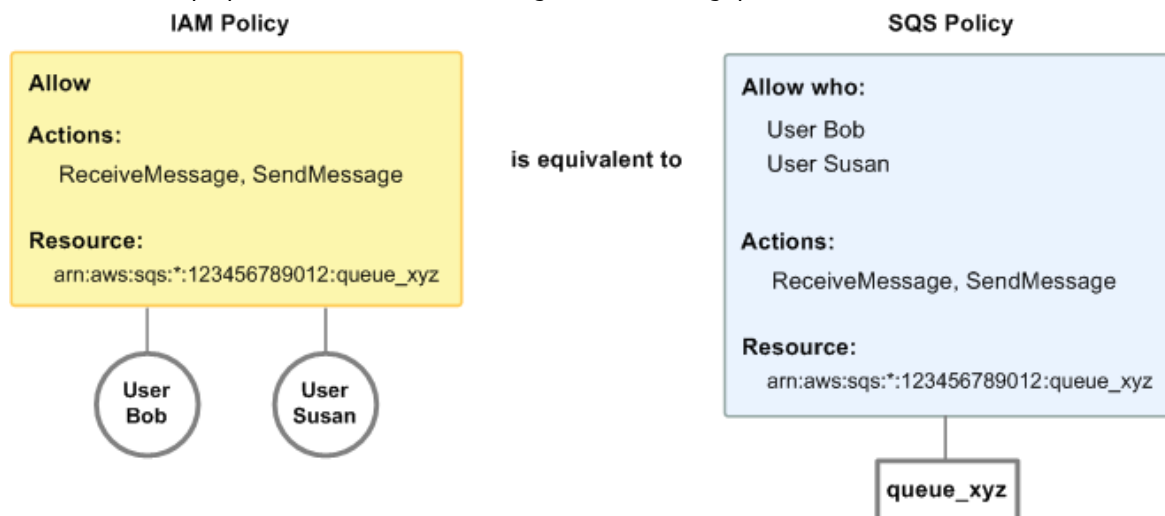
Tópicos

- [Uso de políticas do Amazon SQS e do IAM \(p. 121\)](#)
- [Permissões necessárias para usar o console do Amazon SQS \(p. 123\)](#)
- [Políticas gerenciadas \(predefinidas\) pela AWS para o Amazon SQS \(p. 123\)](#)
- [Exemplos básicos de políticas do IAM para o Amazon SQS \(p. 124\)](#)
- [Exemplos básicos de políticas do Amazon SQS \(p. 126\)](#)

Uso de políticas do Amazon SQS e do IAM

Há duas maneiras de conceder aos usuários permissões aos seus recursos do Amazon SQS: usando o sistema 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 do Amazon SQS equivalentes a ela. A política do IAM concede os direitos ao Amazon SQS `ReceiveMessage` e `SendMessage` para a fila chamada `queue_xyz` na sua conta da AWS, e a política é anexada aos usuários Bob e Susan (Bob e Susan têm as permissões declaradas na política). Essa política do Amazon SQS também concede a Bob e Susan direitos à propriedade do `ReceiveMessage` e `SendMessage` para a mesma fila.



Note

Esse exemplo mostra políticas simples sem condições. Você pode especificar uma determinada condição na política e obter o mesmo resultado.

Há uma diferença importante entre as políticas do IAM e do Amazon SQS: o sistema de política do Amazon SQS permite conceder permissões a outras contas da AWS, enquanto o IAM não permite.

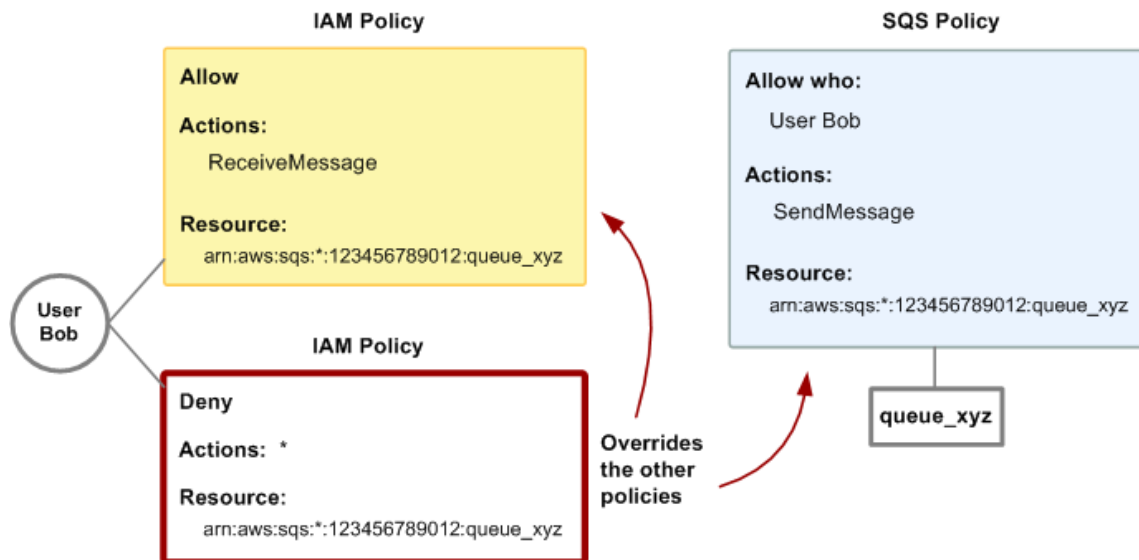
Você é quem decide como usar os dois sistemas para gerenciar suas permissões. Os exemplos a seguir mostram como os dois sistemas de política funcionam em conjunto.

- No primeiro exemplo, Bob tem uma política do IAM e uma política do Amazon SQS que se aplicam à sua conta. A política do IAM concede a sua conta permissão para o `ReceiveMessage` ação em `queue_xyz`, enquanto a política do Amazon SQS dá a sua conta permissão para o `SendMessage` ação na mesma fila. O seguinte diagrama ilustra o conceito.



Se Bob enviar um `ReceiveMessage` Solicitação para `queue_xyz`, a política do IAM permite a ação. Se Bob enviar um `SendMessage` Solicitação para `queue_xyz` A política do Amazon SQS permite 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 [Uso de políticas personalizadas com a linguagem de política de acesso do Amazon SQS](#) (p. 129). 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 abrangem ações e recursos do Amazon SQS, consulte [Exemplos básicos de políticas do Amazon SQS](#) (p. 126). Para obter mais informações sobre como escrever políticas do Amazon SQS, consulte [Uso de políticas personalizadas com a linguagem de política de acesso do Amazon SQS](#) (p. 129).

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 conceder permissões mínimas do console para os usuários que fazem chamadas para as ações da CLI da AWS ou do Amazon SQS.

Políticas gerenciadas (predefinidas) pela AWS para o Amazon SQS

A AWS resolve muitos casos de uso comuns fornecendo políticas do IAM gerenciadas pela AWS. Essas políticas gerenciadas pela AWS simplificam o trabalho com permissões ao atribuir as permissões

necessárias para casos de uso comuns. Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) no Guia do usuário do IAM.

As seguintes políticas gerenciadas pela AWS (que você pode associar a usuários em sua conta) são específicas do Amazon SQS:

- **AmazonSQSReadOnlyAccess**— concede acesso somente leitura a filas do Amazon SQS usando o Console de Gerenciamento da AWS.
- **AmazonSQSFullAccess**— concede pleno acesso às filas do Amazon SQS usando o Console de Gerenciamento da AWS.

Você pode pesquisar e analisar políticas disponíveis no console do IAM. Você também pode criar suas próprias políticas do IAM personalizadas a fim de conceder permissões para ações e filas do Amazon SQS. Você pode anexar essas políticas personalizadas a usuários ou grupos do IAM que exijam permissões.

Exemplos básicos de políticas do IAM para o Amazon SQS

Os exemplos a seguir fornecem uma introdução às políticas de permissão do Amazon SQS.

Note

Ao configurar ganchos do ciclo de vida do Amazon EC2 Auto Scaling, você não precisa criar uma política para enviar mensagens a uma fila do Amazon SQS. Para obter mais informações, consulte [Ganchos de ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Exemplo 1: Permitir que um usuário crie filas

No exemplo a seguir, criamos uma política para Bob que lhe permite acessar todas as ações do Amazon SQS, mas apenas com as filas cujos nomes estejam prefixados com a string literal `alice_queue_`.

O Amazon SQS não concede automaticamente ao criador de uma fila permissões para usar a fila. Portanto, é preciso conceder explicitamente permissões a Bob para usar todas as ações do Amazon SQS além de `CreateQueue` na política do IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "sqs:*",
    "Resource": "arn:aws:sqs:*:123456789012:alice_queue_*"
  }]
}
```

Exemplo 2: Permitir que desenvolvedores gravem mensagens em uma fila compartilhada

No exemplo a seguir, criamos um grupo para desenvolvedores e anexamos uma política que permite que o grupo utilize a do Amazon SQS `SendMessage`, mas somente com a fila que pertence à conta especificada da AWS e é chamada `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*O inclui 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 *.

Exemplo 3: Permitir que 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 utilize a do Amazon SQS GetQueueAttributes Ação com todas as filas que pertencem à conta da AWS especificada.

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

Exemplo 4: 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 o parceiro tiver uma conta da AWS, pode ser mais fácil usar uma política do Amazon SQS. No entanto, qualquer usuário da empresa do parceiro que tenha acesso às credenciais de segurança da AWS pode enviar mensagens para a fila. Para limitar o acesso a um determinado usuário ou aplicativo, 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. Criar um grupo chamado WidgetCo para representar a empresa do parceiro.
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 usar duas janelas do navegador: uma para conceder as permissões e a outra para fazer login no AWS Management Console usando as credenciais do usuário para verificar as permissões à medida que forem concedidas ao usuário.

Exemplo 1: Conceda uma permissão a uma conta da AWS

A política de exemplo a seguir concede número de conta da AWS111122223333oSendMessagepermissão para a fila denominada444455556666/queue1na 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: Conceda duas permissões a uma conta da AWS

A política de exemplo a seguir concede número de conta da AWS111122223333tanto oSendMessageeReceiveMessagepermissão para a fila denominada444455556666/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: Conceda todas as permissões a duas contas da AWS

A política de exemplo a seguir concede dois números de contas diferentes da AWS (111122223333e444455556666) permissão para usar todas as ações às quais o Amazon SQS permite acesso compartilhado para a fila denominada123456789012/queue1na 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: Conceda permissões entre contas a uma função e um nome de usuário

Concessão de políticas de exemplo a seguirrole1eusername1no número da conta da AWS111122223333Permissão entre contas para usar todas as ações às quais o Amazon SQS permite acesso compartilhado para a fila denominada123456789012/queue1na região Leste dos EUA (Ohio).

As permissões entre contas não se aplicam às seguintes ações:

- [AddPermission](#)
- [CreateQueue](#)
- [DeleteQueue](#)
- [ListQueues](#)
- [ListQueueTags](#)
- [RemovePermission](#)
- [SetQueueAttributes](#)
- [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: Concessão de 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: Conceda 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: Conceda todas as permissões a todos os usuários em um intervalo CIDR

O exemplo de política 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 vem do `192.168.143.0/24` Intervalo CIDR.

```
{
  "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": "192.168.143.0/24"
        }
    }
}
}]
}
```

Exemplo 8: Permissões de 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 192.168.143.0/24 CIDR (exceto para 192.168.143.188) permissão para usar a ação `SendMessage` para a fila denominada 111122223333/queue1.
- A segunda instrução impede que todos os usuários (anônimos) no intervalo CIDR 10.1.2.0/24 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": "192.168.143.0/24"
      },
      "NotIpAddress" : {
        "aws:SourceIp": "192.168.143.188/32"
      }
    }
  }, {
    "Sid": "Queue1_AnonymousAccess_AllActions_IPLimit_Deny",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "sqs:*",
    "Resource": "arn:aws:sqs:*:111122223333:queue1",
    "Condition" : {
      "IpAddress" : {
        "aws:SourceIp": "10.1.2.0/24"
      }
    }
  }
}]
}
```

Uso de políticas personalizadas com a linguagem de política de acesso do Amazon SQS

Se você deseja conceder acesso ao Amazon SQS com base apenas em um ID de conta da AWS e permissões básicas (como para [SendMessage](#) ou [ReceiveMessage](#)), você não precisa escrever suas próprias políticas do. Você só pode usar o Amazon SQS [AddPermission](#) Ação .

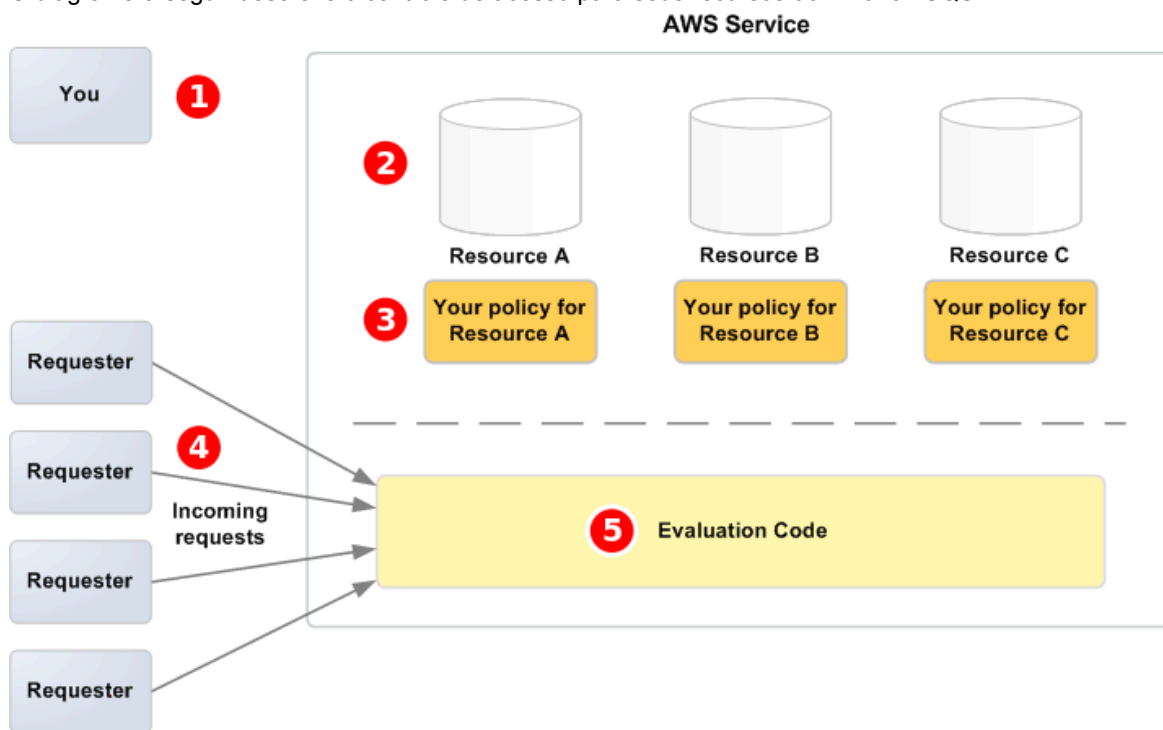
Para negar ou permitir explicitamente o acesso com base em condições mais específicas (como o momento em que a solicitação é recebida ou o endereço IP do solicitante), precisa criar suas próprias políticas do Amazon SQS e obtê-las por upload no sistema da AWS usando o `Amazon SQS SetQueueAttributes` ação.

Tópicos

- [Arquitetura de controle de acesso do Amazon SQS \(p. 130\)](#)
- [Fluxo Amazon SQS processo de controle de acesso do \(p. 131\)](#)
- [Conceitos principais da linguagem da política de acesso do Amazon SQS \(p. 131\)](#)
- [Lógica de avaliação da linguagem da política de acesso do Amazon SQ \(p. 133\)](#)
- [Relações entre negações explícitas e padrão no Amazon SQS Access Policy Language \(p. 135\)](#)
- [Exemplos personalizados de linguagem de política de acesso do Amazon SQS \(p. 137\)](#)

Arquitetura de 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 serviço da AWS (por exemplo, filas do Amazon SQS).

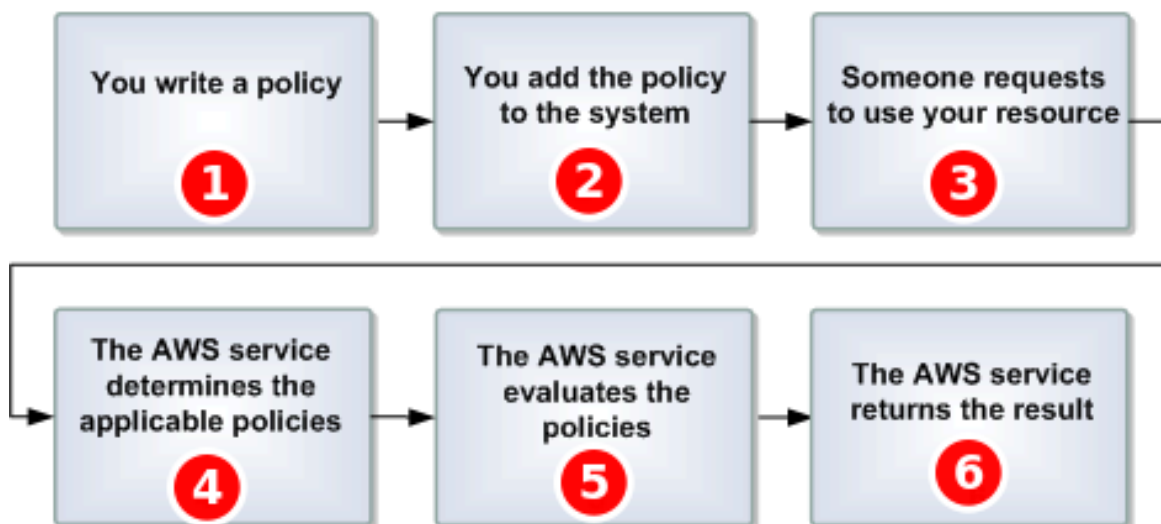
3 Suas políticas. É uma boa prática ter uma política por recurso. O serviço da AWS fornece uma API que você usa para carregar e gerenciar suas políticas.

4 Os solicitantes e suas solicitações de entrada para o serviço da AWS.

5 O código de avaliação da linguagem da política de acesso. Esse é o conjunto de códigos dentro do serviço da AWS que avalia as solicitações de entrada em relação a políticas aplicáveis e determina se o solicitante tem o acesso permitido ao recurso.

Fluxo Amazon SQS processo de controle de acesso do

O diagrama a seguir descreve o fluxo de trabalho geral do controle de acesso com a linguagem da política de acesso do Amazon SQS.



1 Você grava uma política do Amazon SQS para a fila.

2 Você carrega a política na AWS. O serviço da AWS fornece uma API que você usa para fazer upload de suas políticas. Por exemplo, você usa o `Amazon SQS:SetQueueAttributes` Ação do para 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 do disponíveis e determina quais são aplicáveis.

5 O Amazon SQS avalia as políticas do e determina se o solicitante tem permissão para usar sua fila.

6 Com base no resultado da avaliação da política, o Amazon SQS retorna uma `AccessDenied` Erro ao solicitante ou continua a processar a solicitação.

Conceitos principais da linguagem da política 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 [Declaração \(p. 133\)](#) que tenha [Efeito \(p. 132\)](#) definido como `allow`.

Ação

A atividade que o [Principal \(p. 132\)](#) tem permissão para executar, normalmente uma solicitação para a AWS.

Negação padrão

O resultado de uma [Declaração \(p. 133\)](#) que não tem as configurações [Permitir \(p. 131\)](#) e [Negação explícita \(p. 132\)](#).

Condição

Qualquer restrição ou detalhe sobre uma [Permissão \(p. 132\)](#). Condições comuns são relacionadas a data e hora e a endereços IP.

Efeito

O resultado que você deseja que a [Declaração \(p. 133\)](#) de uma [Política \(p. 132\)](#) retorne 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 \(p. 131\)](#), [Permitir \(p. 131\)](#) e [Negação explícita \(p. 132\)](#).

Negação explícita

O resultado de uma [Declaração \(p. 133\)](#) que tenha [Efeito \(p. 132\)](#) 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 \(p. 132\)](#).

Emissor

O usuário que grava uma [Política \(p. 132\)](#) para conceder permissões a um recurso. O emissor, por definição, é sempre o proprietário do recurso. A AWS não permite que usuários do Amazon SQS criem políticas para recursos dos quais não sejam proprietários.

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 \(p. 132\)](#) e uma [Chave \(p. 132\)](#).

Política

O documento que atua como um contêiner para uma ou mais [instruções \(p. 133\)](#).



O Amazon SQS usa a política para determinar se deverá conceder acesso a um usuário para um recurso.

Principal

O usuário que recebe [Permissão \(p. 132\)](#) na [Política \(p. 132\)](#).

Recurso

O objeto ao qual a [Principal \(p. 132\)](#) solicita acesso.

Declaração

A descrição formal de uma única permissão, criada no idioma da política de acesso como parte de uma [Política \(p. 132\)](#) document.

Solicitante

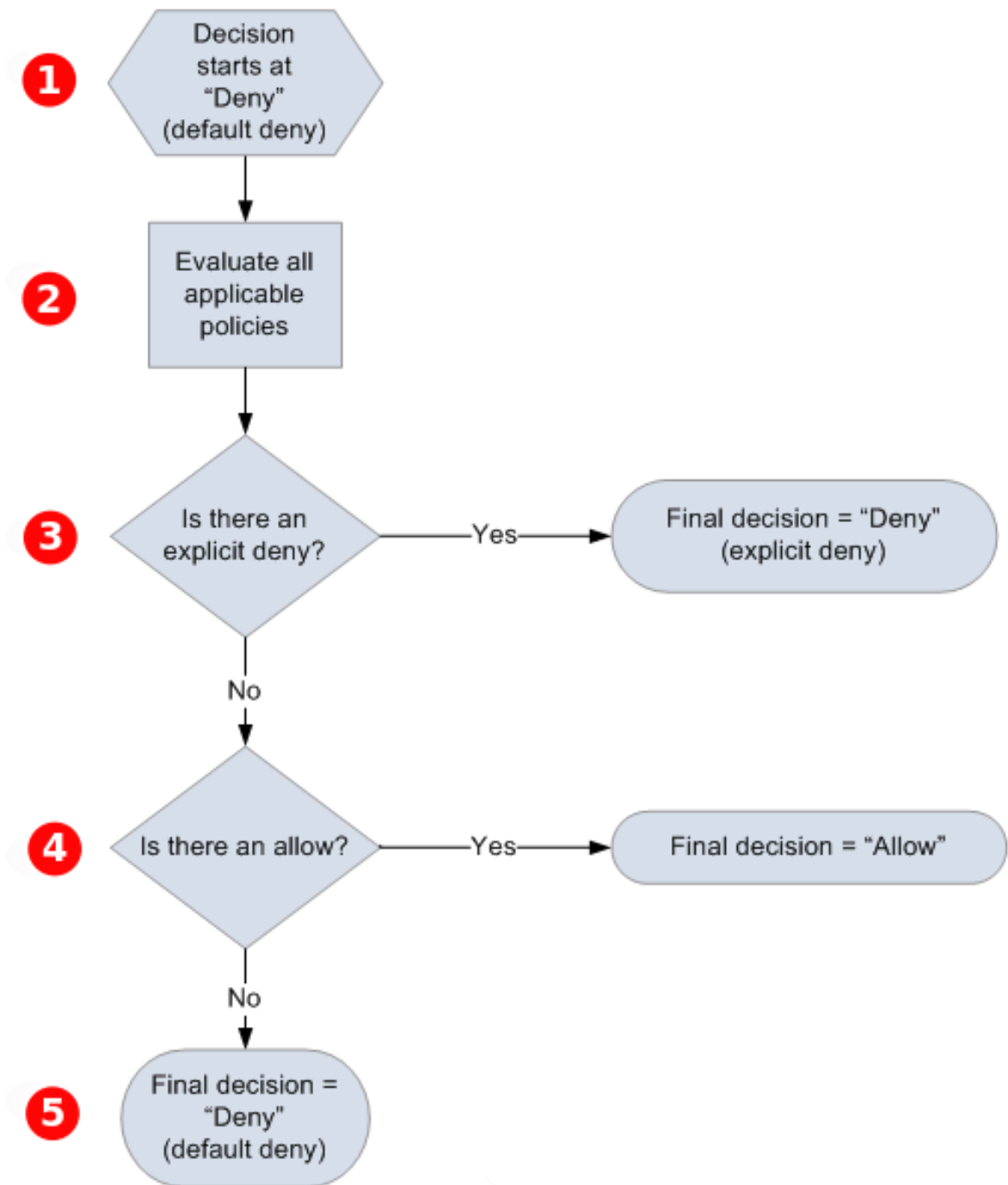
O usuário que envia uma solicitação de acesso a um [Recurso \(p. 132\)](#).

Lógica de avaliação da linguagem da política 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 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 \(p. 131\)](#) substitui qualquer [Negação padrão \(p. 131\)](#).
- Uma [Negação explícita \(p. 132\)](#) 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 decisões sobre permissões de acesso.



- ❶ A decisão começa com uma negação padrão.
- ❷ 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.
- ❸ 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.

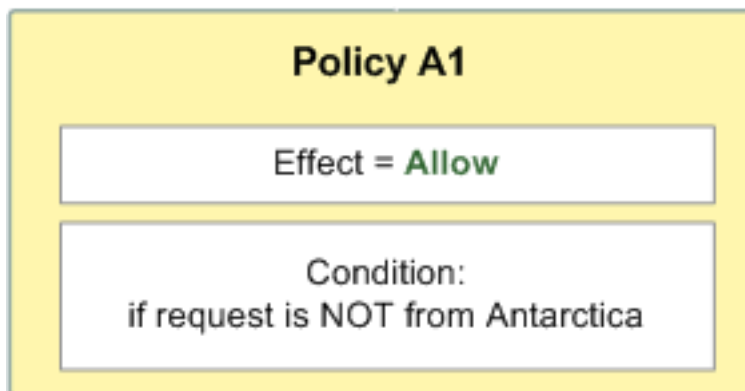
❶ Se nenhuma instrução de negação explícita for encontrada, o código de imposição procurará qualquer instrução de permissão que possa ser aplicada à solicitação. Se encontrar uma instrução "permitir", o código de aplicação retornará uma decisão de permitir, e o processo será concluído (o serviço continua a processar a solicitação).

❷ 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).

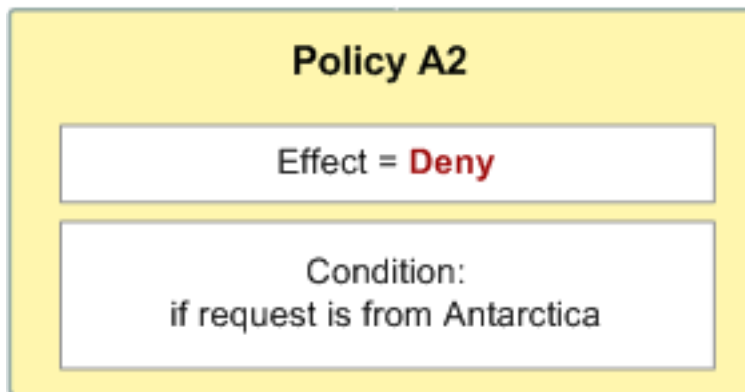
Relações entre negações explícitas e padrão no Amazon SQS Access Policy Language

Se uma política do Amazon SQS não se aplicar diretamente a uma solicitação, a solicitação resultará em uma [Negação padrão](#) (p. 131). Por exemplo, se um usuário solicitar permissão para usar o Amazon SQS, mas a única política que se aplicar ao usuário 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](#) (p. 131) ou [Negação explícita](#) (p. 132) com base no valor do elemento [Efeito](#) (p. 132) 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 deseja evitar solicitações da Antártica. Você elabora uma Política A1 que permite uma solicitação apenas se não vier da Antártica. O diagrama a seguir ilustra a política do Amazon SQS.

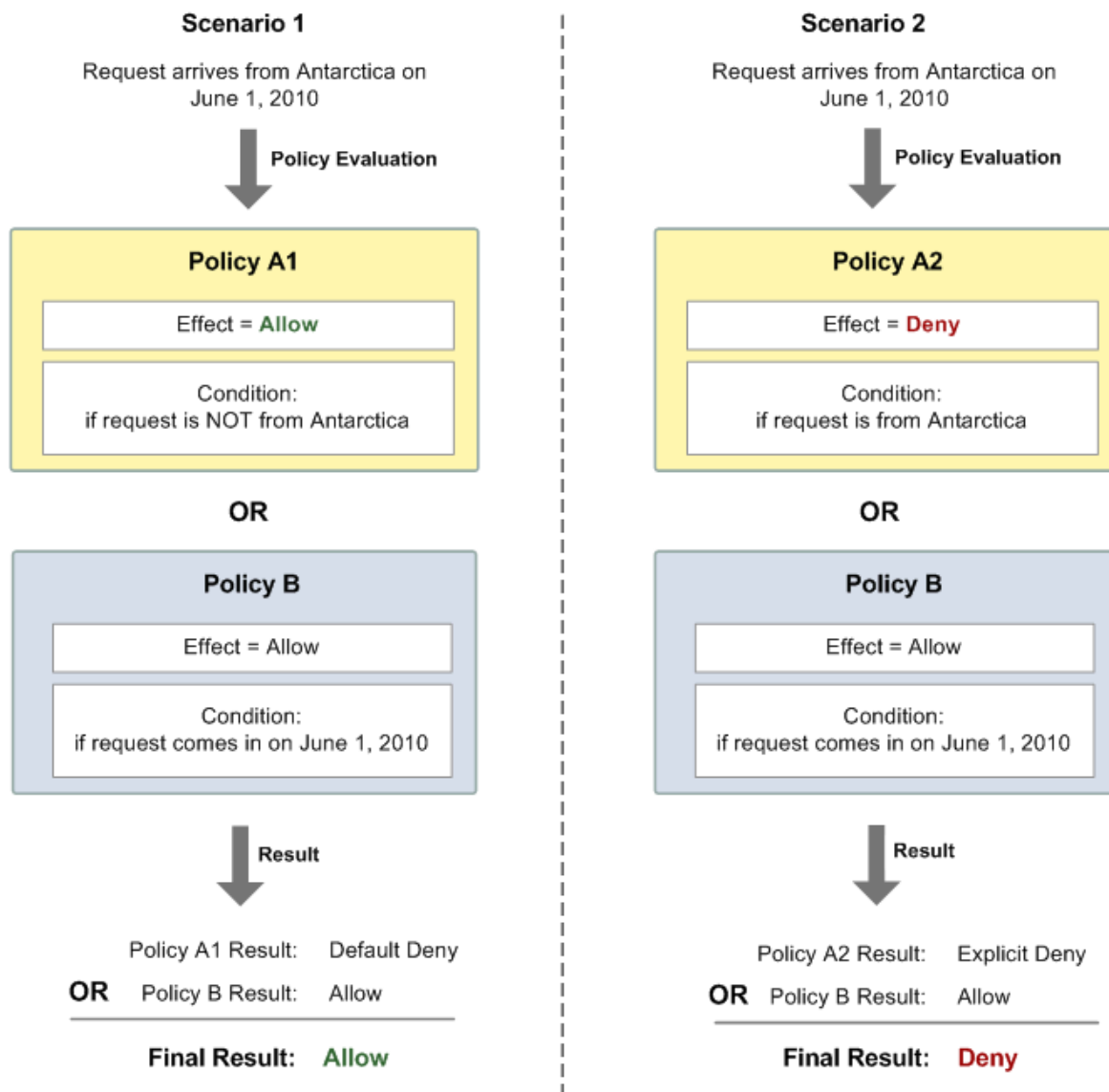


Se um usuário enviar uma solicitação dos EUA, a condição será atendida (a solicitação não será da Antártica) e a solicitação resultará em uma permissão. No entanto, se um usuário enviar uma solicitação da Antártica, a condição não será atendida, e a solicitação será padronizada para uma solicitação padrão. 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.

Exemplos personalizados de linguagem de política de acesso do Amazon SQS

Veja a seguir exemplos de políticas de acesso típicas do Amazon SQS.

Exemplo 1: Concessão de permissão a uma conta

O exemplo de política do Amazon SQS a seguir concede à conta da AWS 111122223333 permissão para enviar para e receber do queue2 de propriedade da conta 444455556666 da AWS.

```
{
  "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: Conceda permissão a uma ou mais contas

O exemplo de política do Amazon SQS a seguir fornece a uma ou mais contas da AWS acesso a filas de propriedade de sua conta por um período específico. É necessário escrever essa política e fazer upload para o Amazon SQS usando o `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: Conceda permissão para solicitações de instâncias do Amazon EC2

O exemplo de política do Amazon SQS a seguir fornece acesso a solicitações provenientes de instâncias do Amazon EC2. Esse exemplo se baseia no exemplo "[Exemplo 2: Conceda permissão a uma ou mais contas \(p. 137\)](#)": 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 escrever essa política e fazer upload para o Amazon SQS usando o `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 política do Amazon SQS a seguir nega a uma conta específica da AWS à sua fila. Este exemplo se baseia no "[Exemplo 1: Concessão de permissão a uma conta \(p. 137\)](#)" exemplo: ele nega o acesso à conta da AWS especificada. É necessário escrever essa política e fazer upload para o Amazon SQS usando o `SetQueueAttributes` porque a ação `AddPermission` não permite negar acesso a uma fila (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 acesso se não vier de um VPC endpoint

O exemplo a seguir a política do Amazon SQS restringe o acesso ao `queue1: 111122223333` pode executar o `SendMessage` e `ReceiveMessage` somente a partir do ID de endpoint da VPC `vpce-1a2b3c4d` (especificado usando `aws:sourceVpce` condição). Para obter mais informações, consulte [Amazon Virtual Private Cloud para o Amazon SQS](#) (p. 113).

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 VPC endpoint negando todas as ações do Amazon SQS (`sqs:*`) na segunda declaração. 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"
      }
    }
  }
]
}
```

Uso de credenciais de segurança temporárias com o Amazon SQS

Além de criar usuários do IAM com suas próprias credenciais de segurança, o IAM também permite conceder credenciais de segurança temporárias a qualquer usuário, possibilitando que o usuário acesse

os serviços e recursos da AWS. Você pode gerenciar os usuários que têm contas da AWS (usuários do IAM). Você também pode gerenciar usuários para seu sistema que não tenham contas da AWS (usuários federados). Além disso, os aplicativos criados para acessar os recursos da AWS também podem ser considerados “usuários”.

Use as 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.

Prerequisites

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 [Conceda acesso temporário aos recursos da AWS](#) 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 o AWS Identity and Access Management. Para obter mais informações, consulte [Criar credenciais 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 de 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 do **AUTHPARAMS** depende da assinatura da solicitação de API. Para obter mais informações, consulte [Como assinar solicitações de API da AWS](#) no 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/bPxrRfiCYEXAMPLEKEY
&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
```

Permissões de API do Amazon SQS: Referência de ações e recursos

Quando você configura [Controle de acesso \(p. 115\)](#) E políticas de permissões de gravação que você pode associar a uma identidade do IAM, você pode usar a tabela a seguir como referência. O A lista inclui Cada ação do Amazon Simple Queue Service, as ações correspondentes às quais você pode conceder permissões para executar a ação e o recurso da AWS ao 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 oferece suporte a um subconjunto limitado de [chaves de condição disponíveis no IAM](#):

- `aws:CurrentTime`
- `aws:EpochTime`
- `aws:SecureTransport`
- `aws:SourceAccount`
- `aws:SourceArn`

Note

Essa condição garante que os serviços da AWS concedam acesso somente em nome dos recursos que sua conta da AWS possui. Você não pode especificar o ARN de uma função do IAM como ARN de origem, porque uma função do IAM não é uma origem nem um serviço.

- `aws:SourceIP`
- `aws:UserAgent`
- `aws:MultiFactorAuthAge`
- `aws:MultiFactorAuthPresent`

- `aws:PrincipalOrgID`
- `aws:RequestTag`
- `aws:sourceVpce`
- `aws:TagKeys`
- `aws:TokenAge`

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`

ChangeMessageVisibility

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`

GetQueueAttributes

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`

ListDeadLetterSourceQueues

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*

ListQueueTags

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*

SendMessage e SendMessageBatch

Ação/Ações: sqs:SendMessage

Recurso: arn:aws:sqs:*region*:*account_id*:*queue_name*

SetQueueAttributes

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 o monitoramento e o registro em filas do Amazon SQS.

Tópicos

- [Registrar chamadas de API Amazon SQS usando o AWS CloudTrail \(p. 144\)](#)
- [Registrar filas do Amazon SQS usando o CloudWatch \(p. 148\)](#)

Registrar chamadas de API Amazon SQS usando o AWS CloudTrail

O Amazon SQS é integrado ao AWS CloudTrail, um serviço que fornece um registro das chamadas do Amazon SQS que um usuário, função ou serviço da AWS faz. O CloudTrail captura chamadas de API relacionadas às filas do Amazon SQS como eventos, incluindo chamadas do console do Amazon SQS e chamadas de código das APIs do Amazon SQS. Para obter mais informações sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

Note

O registro em log do CloudTrail oferece suporte a filas padrão e FIFO.

Usando as informações que o CloudTrail coleta, é possível identificar uma solicitação específica a uma API do Amazon SQS, o endereço IP do solicitante, a identidade do solicitante, a data e a hora da solicitação etc. Se você configurar uma trilha, você pode habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3. Se você não configurar uma trilha, poderá visualizar os eventos mais recentes no histórico de eventos do console do CloudTrail. Para obter mais informações, consulte [Visão geral da criação de uma trilha](#) no [Guia do usuário do AWS CloudTrail](#).

Informações do Amazon SQS no CloudTrail

Quando você cria sua conta da AWS, o CloudTrail é habilitado. Quando a atividade do evento do Amazon SQS com suporte ocorre, ela é registrada em um evento do CloudTrail com outros eventos de serviços da AWS. Você pode visualizar, pesquisar e fazer download de eventos recentes para a sua conta da AWS. Para obter mais informações, consulte [Visualizar eventos com o histórico de eventos do CloudTrail](#) no [Guia do usuário do AWS CloudTrail](#).

Uma trilha permite que o CloudTrail entregue arquivos de log para um bucket do Amazon S3. Você pode criar uma trilha para manter um registro contínuo de eventos em sua conta da AWS. Por padrão, quando você cria uma trilha usando o AWS Management Console, ela é aplicada a todas as regiões da AWS. A trilha registra eventos de todas as regiões da AWS e fornece arquivos de log para o bucket do Amazon S3 especificado. Também é possível configurar outros serviços da AWS para analisar e atuar mais profundamente sobre os dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte os seguintes tópicos no [Guia do usuário do AWS CloudTrail](#):

- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configuração de notificações do Amazon SNS para o CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões](#)
- [Receber arquivos de log do CloudTrail de várias contas](#)

O Amazon SQS oferece suporte a registro em log das seguintes ações:

- [AddPermission](#)
- [CreateQueue](#)
- [DeleteQueue](#)
- [PurgeQueue](#)

- [RemovePermission](#)
- [SetQueueAttributes](#)
- [TagQueue](#)
- [UntagQueue](#)

Cada evento ou entrada de log contém informações sobre o solicitante. As informações ajudam a identificar:

- A solicitação foi feita com credenciais de usuário raiz ou do IAM?
- A solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado?
- A solicitação foi feita por outro serviço da AWS?

Para obter mais informações, consulte [Elemento CloudTrail userIdentity](#) no Guia do usuário do AWS CloudTrail.

Exemplo de entradas de arquivo de log do Amazon SQS

Os arquivos de log do CloudTrail contêm uma ou mais entradas de log, e cada entrada é composta de vários eventos em formato JSON. Uma entrada de log representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, quaisquer parâmetros, a data e hora da ação e assim por diante. Não é garantido que as entradas de log estejam em uma ordem específica. Ou seja, elas não são um rastreamento de pilha ordenada das chamadas de API públicas.

AddPermission

O exemplo a seguir mostra uma entrada de log do CloudTrail para um `AddPermission` Chamada de 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.amazonaws.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 log do CloudTrail para umCreateQueueChamada de 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 log do CloudTrail para umDeleteQueueChamada de 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 log do CloudTrail para um `RemovePermission` Chamada de 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 log do CloudTrail para o `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.amazonaws.com/123456789012/MyQueue"
  },
  "responseElements": null,
  "requestID": "123abcde-f4gh-50ij-klmn-60o789012p30",
  "eventID": "0987g654-32f1-09e8-d765-c4f3fb2109fa"
}
```

Registrar filas do Amazon SQS usando o CloudWatch

O Amazon SQS e o Amazon CloudWatch são integrados, para que você possa usar o CloudWatch para visualizar e analisar métricas para suas filas do Amazon SQS. Você pode visualizar e analisar as métricas das filas no [Console do Amazon SQS \(p. 148\)](#), o [Console do CloudWatch \(p. 149\)](#), usando o [CLI DA AWS \(p. 150\)](#), ou usando o [API do CloudWatch \(p. 150\)](#). Também é possível [Definir alarmes do CloudWatch \(p. 150\)](#) para métricas do Amazon SQS.

As métricas do CloudWatch para suas filas do Amazon SQS são coletadas e enviadas automaticamente para o CloudWatch em intervalos de um minuto. Essas métricas são coletadas em todas as filas que estiverem de acordo com as diretrizes do CloudWatch para serem ativo. O CloudWatch considera uma fila ativa por até seis horas se contiver mensagens ou se qualquer ação de acessá-la.

Note

- Não há cobrança para as métricas do Amazon SQS informadas no CloudWatch. Elas são fornecidas como parte do serviço Amazon SQS.
- As métricas do CloudWatch são compatíveis com filas padrão e FIFO.

Tópicos

- [Acessar CloudWatch do Amazon SQS \(p. 148\)](#)
- [Criação de alarmes do CloudWatch para métricas do Amazon SQS \(p. 150\)](#)
- [Métricas CloudWatch Amazon SQS \(p. 151\)](#)


Acessar CloudWatch do Amazon SQS

O Amazon SQS e o Amazon CloudWatch são integrados, para que você possa usar o CloudWatch para visualizar e analisar métricas para suas filas do Amazon SQS. Você pode visualizar e analisar as métricas das filas no [Console do Amazon SQS \(p. 148\)](#), o [Console do CloudWatch \(p. 149\)](#), usando o [CLI DA AWS \(p. 150\)](#), ou usando o [API do CloudWatch \(p. 150\)](#). Também é possível [Definir alarmes do CloudWatch \(p. 150\)](#) para métricas do Amazon SQS.

Console do Amazon SQS

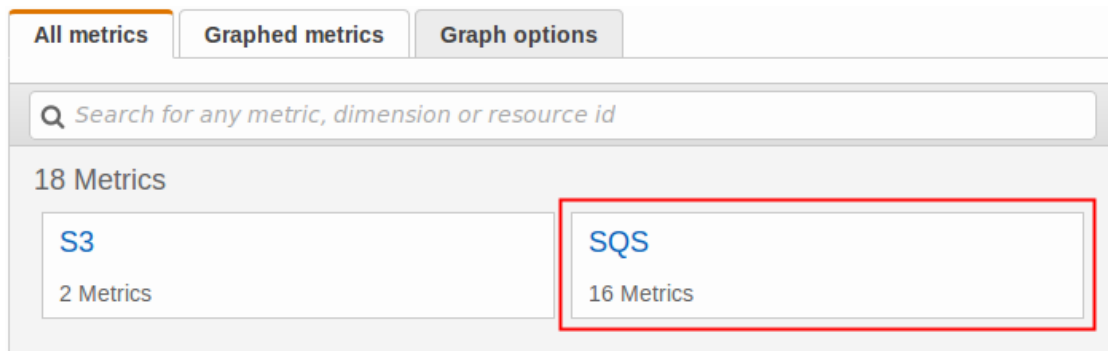
1. Faça login no [Console do Amazon SQS](#).
2. Na lista de filas, escolha (selecione) as caixas das filas cujas métricas você deseja acessar. Você pode exibir métricas para até 10 filas.
3. Escolha a guia Monitoring (Monitoramento).

Vários gráficos são exibidos na seção SQS metrics.

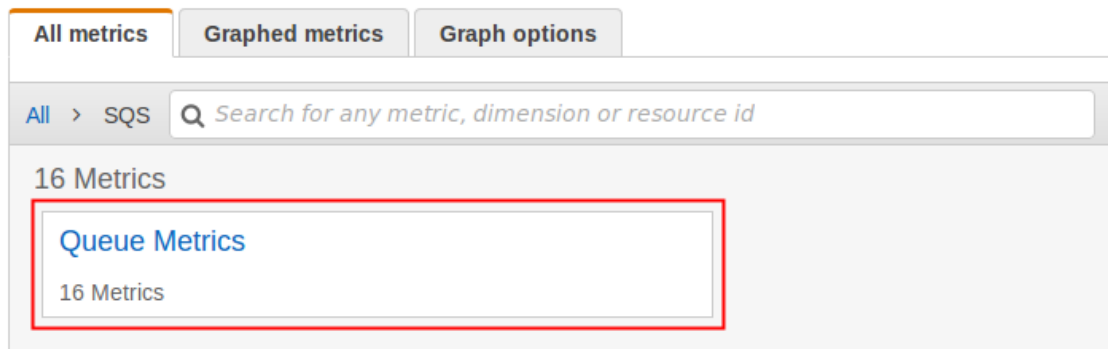
4. Para entender o que um determinado gráfico representa, passe o mouse sobre  ao lado do gráfico desejado, ou consulte [Métricas CloudWatch Amazon SQS \(p. 151\)](#).
5. Para alterar o intervalo de tempo para todos os gráficos ao mesmo tempo, em Time Range, escolha o intervalo de tempo desejado (por exemplo, Last Hour).
6. Para exibir estatísticas adicionais para um gráfico individual, selecione o gráfico.
7. No Detalhes de monitoramento do CloudWatch, selecione uma estatística, (por exemplo, Soma). Para obter uma lista de estatísticas suportadas, consulte [Métricas CloudWatch Amazon SQS \(p. 151\)](#).
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 recursos adicionais do CloudWatch, no Monitoramento, selecione Visualizar todas as métricas do CloudWatch e siga as instruções no [Console do Amazon CloudWatch \(p. 149\)](#) Procedimento do.

Console do Amazon CloudWatch

1. Faça login no [Console do CloudWatch](#).
2. No painel de navegação, selecione Métricas.
3. Selecione o namespace de métrica do SQS.

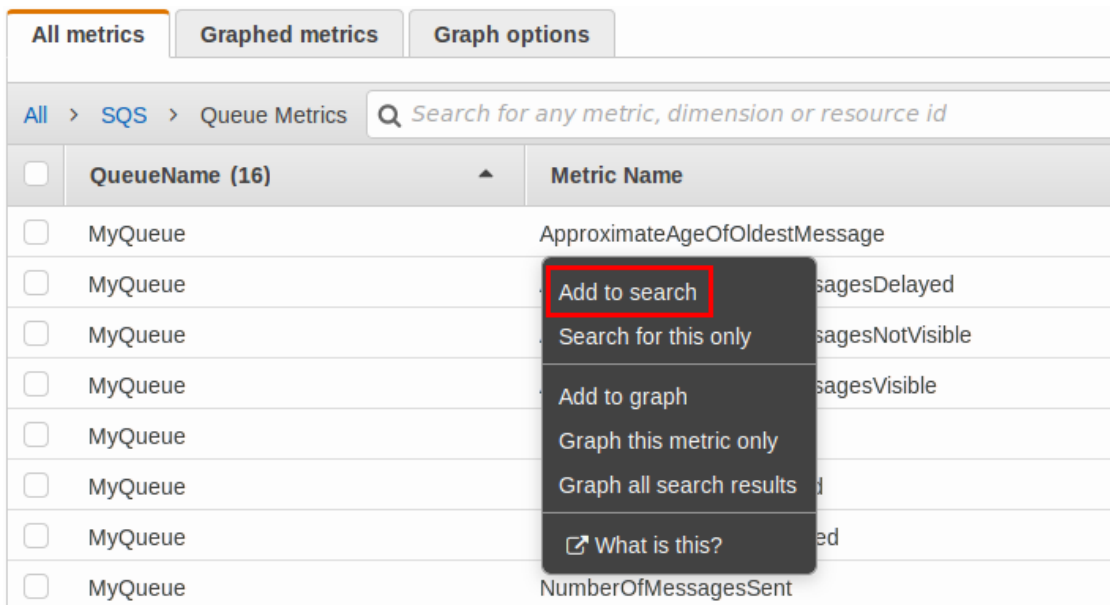


4. Selecione a dimensão de métrica Queue Metrics.



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 Add to search (Adicionar à pesquisa).



Para obter mais informações e opções, consulte [Represente métricas em gráficos](#) e [Usar painéis do Amazon CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

Interface de linha de comando da AWS

Para acessar as métricas do Amazon SQS usando a AWS CLI, execute o `get-metric-statistics` Comando da.

Para obter mais informações, consulte [Obter estatísticas de uma métrica](#) no Guia do usuário do Amazon CloudWatch.

API do CloudWatch

Para acessar as métricas do Amazon SQS usando a API do CloudWatch, use o `GetMetricStatistics` Ação .

Para obter mais informações, consulte [Obter estatísticas de uma métrica](#) no Guia do usuário do Amazon CloudWatch.

Criação de alarmes do CloudWatch para métricas do Amazon SQS

O CloudWatch permite que você acione alertas com base em um limite de métrica. 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 [Criar alarmes do Amazon CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

1. Faça login no Console de Gerenciamento da AWS e abra o console do CloudWatch em <https://console.aws.amazon.com/cloudwatch/>.
2. Escolha Alarms e, em seguida, Create Alarm.

3. Na seção Select Metric (Selecionar métrica) da caixa de diálogo Create Alarm (Criar alerta), selecione Browse Metrics (Procurar métricas), SQS.
4. Em SQS > Queue Metrics (SQS > Métricas de fila), selecione QueueName (Nome da fila) e Metric Name (Nome da métrica) para o qual definir um alerta e, em seguida, selecione Next (Avançar). Para ver uma lista das métricas disponíveis, consulte [Métricas CloudWatch Amazon SQS \(p. 151\)](#).

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 deseja que o CloudWatch envie uma notificação quando o alarme for acionado, selecione um tópico do Amazon SNS existente ou selecione Nova lista e insira os endereços de e-mail separados por vírgula.

Note

Se você criar um novo tópico do Amazon SNS, o endereço 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.

Métricas CloudWatch Amazon SQS

O Amazon SQS envia as seguintes métricas para o CloudWatch.

Note

Para filas padrão, o resultado é aproximado devido à arquitetura distribuída do Amazon SQS. Na maioria dos casos, a contagem deve estar próxima do número 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
<code>ApproximateAgeOfOldestMessage</code>	A idade aproximada de mensagem não excluída mais velha na fila. Note <ul style="list-style-type: none">Depois que uma mensagem é recebida três vezes (ou mais) e não é processada,

Métrica	Descrição
	<p>a mensagem é movida para o final da fila e a métrica <code>ApproximateAgeOfOldestMessage</code> 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.</p> <ul style="list-style-type: none"> • 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 (p. 41) após o número máximo configurado de recebimentos. Quando a mensagem é movida para a dead-letter queue, a métrica <code>ApproximateAgeOfOldestMessage</code> 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). <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Segundos</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p>

Métrica	Descrição
<code>ApproximateNumberOfMessagesDelayed</code>	<p>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.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Contagem</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p>
<code>ApproximateNumberOfMessagesNotVisible</code>	<p>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.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Contagem</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p>
<code>ApproximateNumberOfMessagesVisible</code>	<p>O número de mensagens disponíveis para recuperação da fila.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Contagem</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p>

Métrica	Descrição
<code>NumberOfEmptyReceives¹</code>	<p>O número de chamadas de API <code>ReceiveMessage</code> que não retornaram uma mensagem.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Contagem</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p>
<code>NumberOfMessagesDeleted¹</code>	<p>O número de mensagens excluídas da fila.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Contagem</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p> <p>O Amazon SQS emite <code>NumberOfMessagesDeleted</code> para cada operação bem-sucedida de exclusão usando uma receipt handle, incluindo exclusões duplicadas. Os seguintes cenários podem fazer com que o valor da métrica <code>NumberOfMessagesDeleted</code> seja superior ao esperado:</p> <ul style="list-style-type: none"> Chame <code>deleteMessage</code> em diferentes identificadores de recebimento que pertencem à mesma mensagem: Se a mensagem não for processada antes do Tempo limite de visibilidade do O expirar, a mensagem se torna disponível para outros clientes que podem processá-la e excluí-la novamente, aumentando o valor do <code>NumberOfMessagesDeleted</code> Métrica do. Chame <code>deleteMessage</code> no mesmo identificador de recebimento: Se a mensagem for processada e excluída, mas você chamar o método <code>deleteMessage</code> acionando novamente usando o mesmo identificador de recebimento, um status de sucesso será retornado, aumentando o valor do <code>NumberOfMessagesDeleted</code> Métrica do.

Métrica	Descrição
<code>NumberOfMessagesReceived</code> ¹	<p>O número de mensagens retornadas por chamadas para a ação de <code>ReceiveMessage</code>.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Contagem</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p>
<code>NumberOfMessagesSent</code> ¹	<p>O número de mensagens adicionadas a uma fila.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Contagem</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p>
<code>SentMessageSize</code> ¹	<p>O tamanho das mensagens adicionadas a uma fila.</p> <p>Critérios de relatório Um valor não negativo é relatado se a fila estiver ativa (p. 148).</p> <p>Unidades: Bytes</p> <p>Estatísticas válidas: média, mínimo, máximo, máximo, máximo, soma, amostragens de dados (exibidas como contagem de amostra no console do Amazon SQS)</p> <p>Note</p> <p><code>SentMessageSizeO</code> não é exibido como uma métrica disponível no console do CloudWatch até que pelo menos uma mensagem seja enviada para a fila correspondente.</p>

¹ Essas métricas são calculadas a partir 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 enviada pelo Amazon SQS para o CloudWatch é `QueueName`. Isso significa que todas as estatísticas disponíveis são filtradas por `QueueName`.

Validação de conformidade do Amazon SQS

Audidores terceiros avaliam a segurança e a conformidade do Amazon SQS como parte de vários programas de conformidade da AWS, incluindo o seguinte:

- Padrão de segurança de dados do setor de cartão de pagamento (Payment Card Industry Data Security Standard, PCI DSS)
- Lei de Portabilidade e Responsabilidade de Provedores de Saúde (HIPAA) dos EUA

Para obter uma lista de serviços da AWS no escopo de programas de conformidade específicos, consulte [Serviços da AWS no escopo por programa de conformidade](#). Para obter informações gerais, consulte [Programas de conformidade da AWS](#).

É possível fazer download de relatórios de auditoria externa usando o AWS Artifact. Para obter mais informações, consulte [Fazer download de relatórios no AWS Artifact](#).

Sua responsabilidade de conformidade ao usar o Amazon SQS é determinada pela confidencialidade dos dados, pelos objetivos de conformidade da empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- [Security and Compliance Quick Start Guides](#): esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base concentrados em conformidade e segurança na AWS.
- [Whitepaper Arquitetura para segurança e conformidade com a HIPAA](#): esse whitepaper descreve como as empresas podem usar a AWS para criar aplicações em conformidade com a HIPAA.
- [Recursos de conformidade da AWS](#): essa coleção de manuais e guias pode ser aplicada ao seu setor e seu local.
- [Avaliar recursos com regras](#) no Guia do desenvolvedor do AWS Config: o serviço AWS Config avalia como as configurações de recursos estão em conformidade com práticas internas, diretrizes do setor e regulamentos.
- [AWS Security Hub](#): este serviço da AWS fornece um panorama abrangente do seu estado de segurança dentro da AWS, ajudando você a verificar sua conformidade com os padrões e as práticas recomendadas do setor de segurança.

Resiliência no Amazon SQS

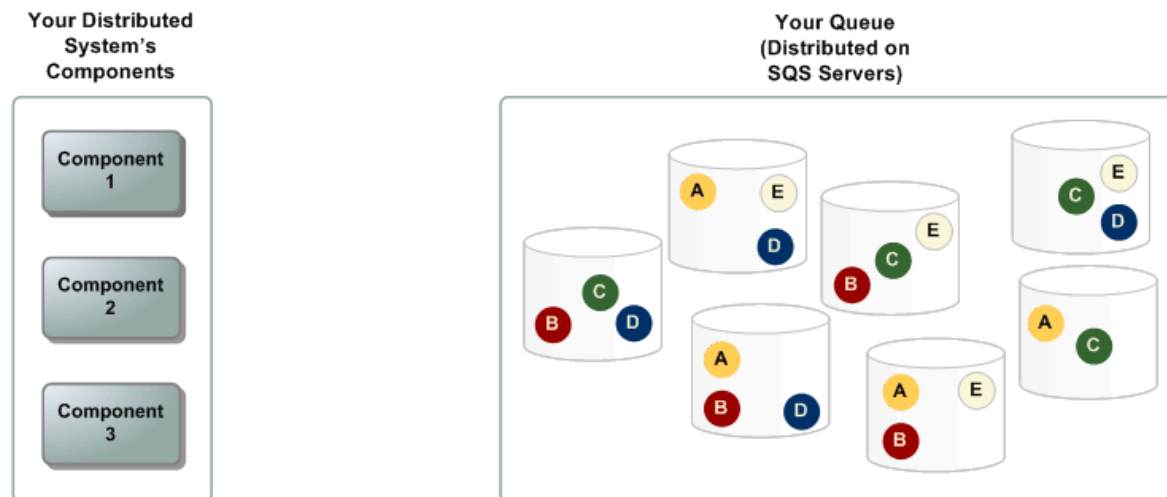
A infraestrutura global da AWS é criada com base em regiões e zonas de disponibilidade da AWS. As regiões da AWS fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, altas taxas de transferência e redes altamente redundantes. Com as zonas de disponibilidade, você pode projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais. Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte [Infraestrutura global da AWS](#).

Além da infraestrutura global da AWS, 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, seu sistema tem vários produtores do (componentes que enviam mensagens para a fila) e Consumidores (componentes que recebem mensagens da fila). A fila (que contém as mensagens de A a E) armazena as mensagens de forma redundante em vários servidores do Amazon SQS.



Segurança da infraestrutura no Amazon S3

Como serviço gerenciado, o Amazon SQS é protegido pelos procedimentos de segurança da rede global da AWS descritos no [Amazon Web Services: Visão geral dos processos de segurança](#) Whitepaper.

Você usa as ações de API da AWS para acessar o Amazon SQS por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos 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 solicitações usando um ID da chave de acesso e uma chave de acesso secreta associados a uma entidade principal do IAM. Você também pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para solicitações de assinatura.

É possível chamar essas ações de API de qualquer local da rede, mas o Amazon SQS oferece suporte a políticas de acesso baseadas 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 Amazon VPC endpoints específicos ou de VPCs específicas. Isso isola efetivamente o acesso à rede para determinada fila do Amazon SQS somente da VPC específica dentro da rede da AWS. Para obter mais informações, consulte [Exemplo 5: Negar acesso se não vier de um VPC endpoint](#) (p. 139).

Melhores práticas de segurança do Amazon SQS

A AWS fornece muitos recursos de segurança para o Amazon SQS, que você deve analisar no contexto da sua própria política de segurança.

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.

Melhores práticas preventivas

Veja a seguir as melhores práticas de segurança preventivas do para o Amazon SQS.

Tópicos

- [Verifique se as filas não estão acessíveis ao público geral \(p. 158\)](#)
- [Implemente o privilégio de acesso mínimo \(p. 158\)](#)
- [Usar funções do IAM para aplicativos e serviços da AWS que exigem acesso ao Amazon SQS \(p. 158\)](#)
- [Implemente a criptografia no lado do servidor \(p. 159\)](#)
- [Aplique a criptografia de dados em trânsito \(p. 159\)](#)
- [Considere usar VPC endpoints para acessar o Amazon SQS \(p. 159\)](#)

Verifique se as filas não estão acessíveis ao público geral

A menos que seja nitidamente necessário que qualquer pessoa na Internet possa ler ou escrever na sua fila do Amazon SQS, você deve garantir que a fila não seja acessível ao público (acessível por todo mundo ou por qualquer usuário autenticado da 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 usa 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 do— acesso para enviar mensagens às filas.
- Consumidores do— acesso para receber e excluir mensagens nas filas.

Para obter mais informações, consulte as seções a seguir:

- [Identity and Access Management no Amazon SQS \(p. 114\)](#)
- [Permissões de API do Amazon SQS: Referência de ações e recursos \(p. 141\)](#)
- [Uso de políticas personalizadas com a linguagem de política de acesso do Amazon SQS \(p. 129\)](#)

Usar funções do IAM para aplicativos e serviços da AWS que exigem acesso ao Amazon SQS

Para que aplicações ou serviços da AWS como o Amazon EC2 acessem filas do Amazon SQS, eles devem usar credenciais válidas da AWS em suas solicitações de API da AWS. Como essas credenciais

não são alteradas automaticamente, você não deve armazenar credenciais da AWS diretamente no aplicativo ou na instância do EC2.

Use uma função do IAM para gerenciar credenciais temporárias para aplicativos ou serviços que precisam de acesso ao Amazon SQS. Quando você usa uma função, não é necessário distribuir credenciais de longo prazo (como um nome de usuário, uma senha e chaves de acesso) para uma instância do EC2 ou serviço da AWS como o AWS Lambda. Em vez disso, a função fornece permissões temporárias que os aplicativos podem usar ao fazer chamadas para outros recursos da AWS.

Para obter mais informações, consulte [Funções do IAM](#) e [Cenários comuns para funções: Utilizadores, aplicações e serviços](#) 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 de 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. O SSE usa chaves gerenciadas pelo 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 \(p. 106\)](#) e [Gerenciamento de chaves \(p. 108\)](#).

Aplique a criptografia de dados em trânsito

Sem HTTPS (TLS), um invasor que venha pela rede pode bisbilhotar o tráfego de rede ou manipulá-lo usando um ataque, como 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 VPC endpoints 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 às filas de endpoints da Amazon VPC ou VPCs específicas.

Os VPC endpoints 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 obter mais informações, consulte [Amazon Virtual Private Cloud para o Amazon SQS \(p. 113\)](#) e [Criar uma política de VPC endpoint para o Amazon SQS \(p. 113\)](#).

Trabalhando com APIs do Amazon SQS

Esta seção fornece informações sobre a criação de endpoints do Amazon SQS, criando solicitações de API de consulta com `GET` e `POST` usando ações de API de lote. Para obter informações detalhadas sobre o Amazon SQS [Ações do](#)—incluindo parâmetros, erros, exemplos e [Tipos de dados do](#)—consulte o [Referência da API do Amazon Simple Queue Service](#).

Para acessar o Amazon SQS usando várias linguagens de programação, você também pode usar o [SDKs da AWS](#) que contêm a seguinte funcionalidade automática:

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

Para obter informações sobre a ferramenta de linha de comando, consulte as seções do Amazon SQS na seção [Referência de comandos da CLI da AWS](#) e a [Referência do cmdlet da AWS Tools for PowerShell](#).

Tópicos

- [Fazer solicitações da API de consulta](#) (p. 160)
- [Ações de lote do Amazon SQS](#) (p. 166)

Fazer solicitações da API de consulta

Nesta seção, você aprende a criar um endpoint do Amazon SQS, fazendo com que o `GET` e `POST` solicite e interprete respostas.

Tópicos

- [Criar um endpoint](#) (p. 160)
- [Como fazer uma solicitação GET](#) (p. 161)
- [Como fazer uma solicitação POST](#) (p. 161)
- [Autenticação de solicitações](#) (p. 162)
- [Interpretação de respostas](#) (p. 165)

Criar um endpoint

Para trabalhar com filas do Amazon SQS, você precisa criar um endpoint. Para obter informações sobre endpoints do Amazon SQS, consulte as páginas a seguir na [Referência geral do Amazon Web Services](#):

- [Endpoints regionais](#)
- [Endpoints e cotas do Amazon Simple Queue Service](#)

Todos os endpoints do Amazon SQS são independentes. Por exemplo, se duas filas forem nomeadas `MyQueue`, uma tiver um endpoint `sqs.us-east-2.amazonaws.com` e a outra tiver o endpoint `sqs.eu-west-2.amazonaws.com`, elas não compartilham 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 de filas e URLs de fila diferenciam maiúsculas e minúsculas.

A estrutura do `AUTHPARAMS` depende da assinatura da solicitação de API. Para obter mais informações, consulte [Como assinar solicitações de API da AWS](#) no Referência geral do Amazon Web Services.

Como fazer uma solicitação GET

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

- **Endpoint**— O recurso no qual a solicitação está agindo (o [Nome e URL da fila \(p. 33\)](#)), por exemplo: `https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue`
- **Ação**— O [Ação](#) do que você quer executar no endpoint. Um ponto de interrogação (?) separa o endpoint da ação, por exemplo: `?Action=SendMessage&MessageBody=Your%20Message%20Text`
- **Parâmetros**— Todos os parâmetros de solicitação — cada parâmetro é separado por um E comercial (&), por exemplo: `&Version=2012-11-05&AUTHPARAMS`

Veja a seguir um exemplo de um `GET` Solicitação que envia uma mensagem para 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 de filas e 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 resto do exemplo não foi codificado no URL para facilitar a leitura.)

Como fazer uma solicitação POST

Um Amazon SQS `POST` solicitação envia parâmetros de consulta como um formulário no corpo de uma solicitação HTTP.

Veja a seguir o exemplo de um 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 acompanhado de um `form-urlencoded` POST Solicitação que envia uma mensagem para 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. `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.

Autenticação de solicitações

A autenticação é o processo para identificar e verificar quem envia uma solicitação. Durante o primeiro estágio de autenticação, a AWS verifica a identidade do produtor e se o produtor está [registrado para usar a AWS](#) (para obter mais informações, consulte [Etapa 1: Criar uma conta da AWS \(p. 4\)](#) e [Etapa 2: Criar um usuário do IAM \(p. 4\)](#)). Em seguida, a AWS segue este 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 \(p. 162\)](#)
- [Parte 1: A solicitação do \(p. 163\)](#)
- [Parte 2: A resposta da AWS \(p. 164\)](#)

Processo de autenticação básica com HMAC-SHA

Ao acessar o Amazon SQS usando API de consulta, você precisa informar o seguinte para autenticar sua solicitação:

- OID de chave de acesso da AWS que identifica sua conta da AWS, que a AWS usa para pesquisar sua chave de acesso secreta.
- O Assinatura de solicitação HMAC-SHA, que é calculado 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 é compatível com o Signature versão 4, que fornece segurança e desempenho aprimorados com base em SHA256 nas versões anteriores. Ao criar novos aplicativos que usam o Amazon SQS, você deve 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).

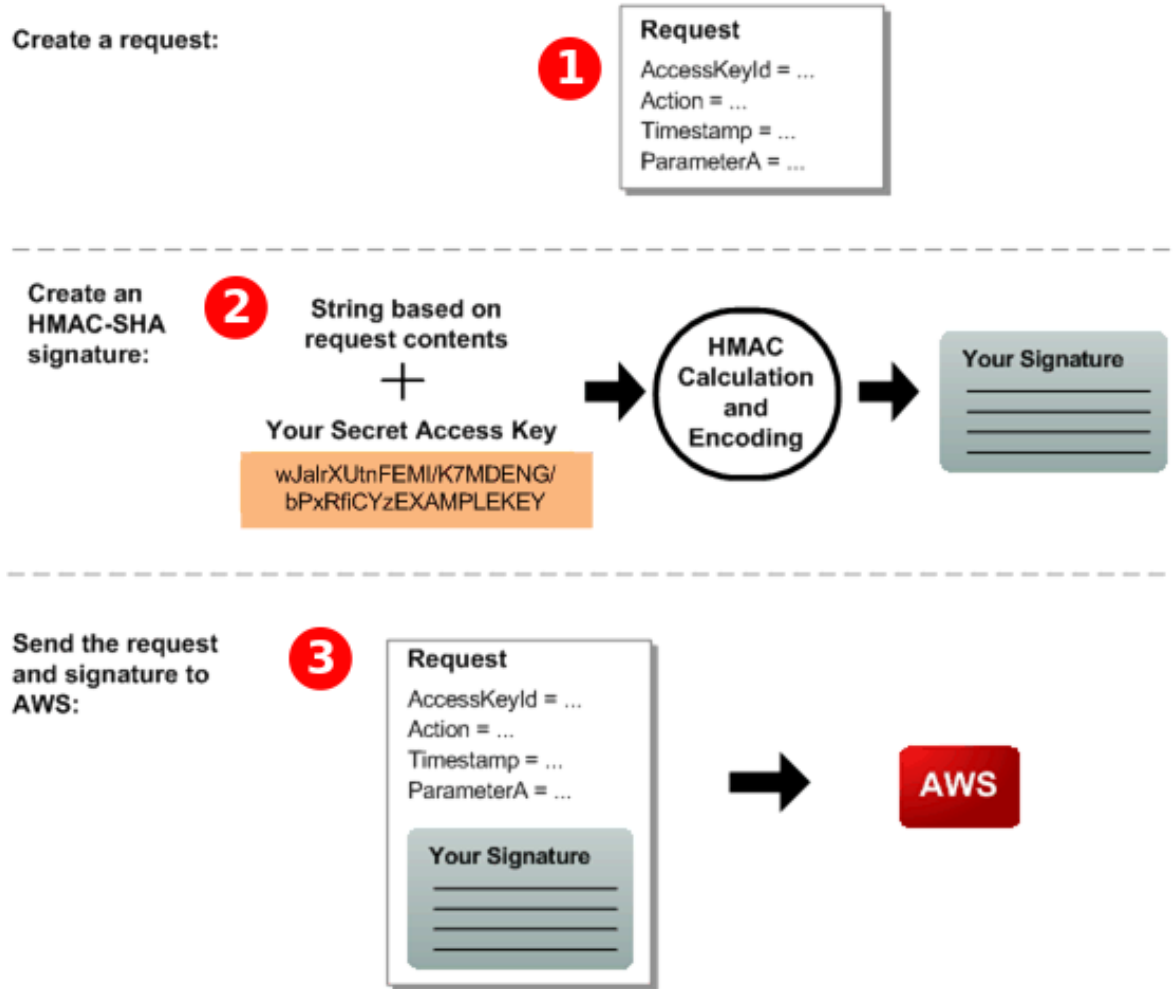
Note

Certifique-se de que a hora do servidor esteja definida corretamente. Se você especificar um timestamp (em vez de uma expiração), a solicitação expirará automaticamente 15 minutos após o horário especificado (a AWS não processará solicitações com timestamps superiores a 15 minutos antes da hora atual nos servidores da 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

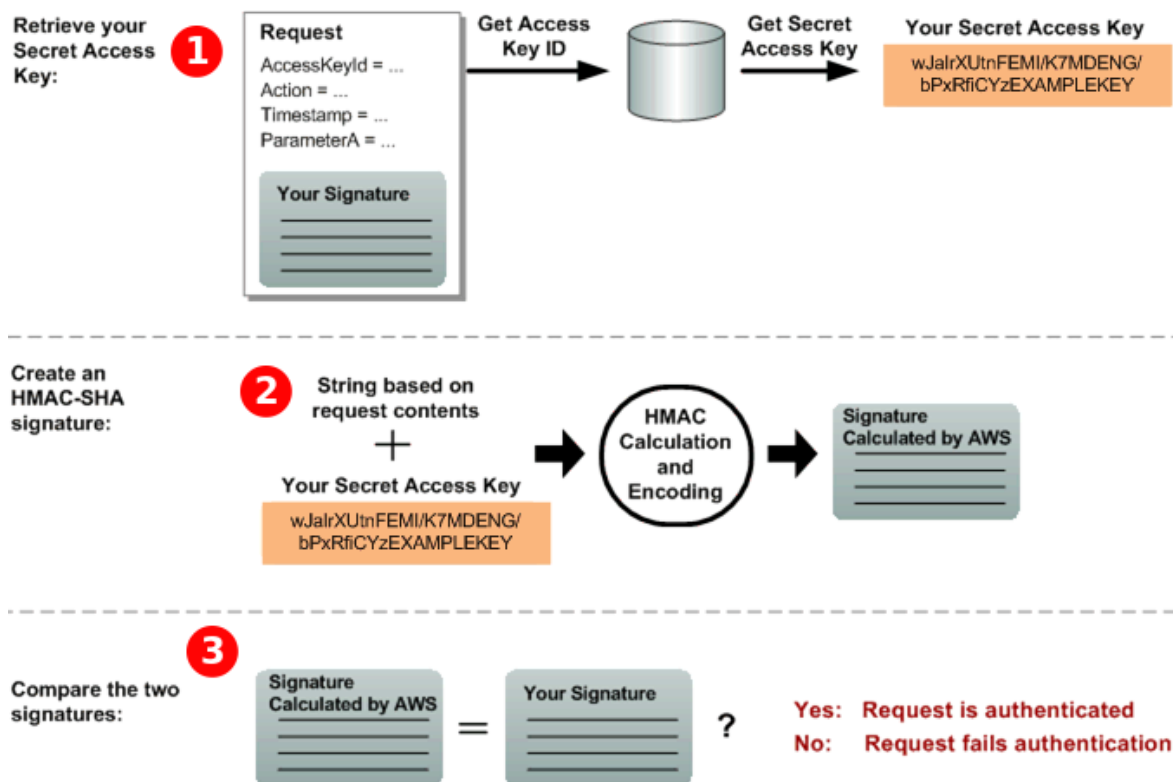
Veja a seguir o processo que você precisa seguir para autenticar solicitações da AWS usando uma assinatura de solicitação HMAC-SHA.



1. Crie uma solicitação para a 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 a AWS.

Parte 2: A resposta da AWS

A AWS inicia este processo em resposta.



1. A AWS usa o ID de chave de acesso para pesquisar sua chave de acesso secreta.
2. A 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 gerada pela AWS corresponder àquela enviada na solicitação, a AWS considerará a solicitação como autêntica.
 - Se a comparação falhar, a solicitação será descartada e a AWS retornará um erro.

Interpretação de respostas

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 as ações individuais na [Referência da API do Amazon Simple Queue Service](#).

Tópicos

- [Estrutura de resposta bem-sucedida](#) (p. 165)
- [Estrutura de resposta de erro](#) (p. 166)

Estrutura de resposta bem-sucedida

Se a solicitação foi bem-sucedida, o elemento de resposta principal receberá o nome da ação, mas com Response anexada (*ActionNameResponse*).

Esse elemento contém os seguintes elementos filho:

- **ActionNameResult**— 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 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

Se uma solicitação apresentar falha, o Amazon SQS retornará o elemento de resposta `principalErrorResponse`. 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 de 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>
```

Ações de 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](#)

Aproveite a funcionalidade de lote usando a API de consulta ou um AWS SDK que oferece suporte às novas ações de lote do Amazon SQS.

Note

O tamanho total de todas as mensagens que você envia em uma única chamada `SendMessageBatch` não pode exceder 262.144 bytes (256 KB).

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 lotes.

Tópicos

- [Habilitar o buffer no lado do cliente e o processamento de solicitações em lotes](#) (p. 167)
- [Aumento da taxa de transferência usando escalabilidade horizontal e processamento de ações em lotes](#) (p. 171)

Habilitar o buffer no lado do cliente e o processamento de solicitações em lotes

O [AWS SDK para Java](#) inclui `AmazonSQSBufferedAsyncClient` que acessa o Amazon SQS. Esse cliente facilita a criação de lotes de solicitação, usando a ativação de buffer do lado do cliente—as chamadas feitas pelo cliente são armazenadas primeiramente em buffer e enviadas como uma solicitação em lotes para o Amazon SQS.

O armazenamento em buffer do lado do cliente permite que até 10 solicitações sejam armazenadas em buffer e enviadas em lote, diminuindo o custo de usar o Amazon SQS e reduzindo a quantidade de solicitações enviadas. `AmazonSQSBufferedAsyncClient` O armazena em buffer chamadas síncronas e assíncronas. Solicitações em lote e suporte para [sondagem longa](#) (p. 39) também podem ajudar a aumentar a taxa de transferência. Para obter mais informações, consulte [Aumento da taxa de transferência usando escalabilidade horizontal e processamento de ações em lotes](#) (p. 171).

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

- [Usar o AmazonSQSBufferedAsyncClient](#) (p. 167)
- [Configurar o AmazonSQSBufferedAsyncClient](#) (p. 168)

Usar o AmazonSQSBufferedAsyncClient

Antes de começar, conclua as tarefas em [Configurar o Amazon SQS](#) (p. 4).

Important

O AWS SDK para Java 2.x não é compatível com o `AmazonSQSBufferedAsyncClient`.

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 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.setRequestBody( body );
request.setQueueUrl(res.getQueueUrl());

final SendMessageResult sendResult = bufferedSqs.sendMessageAsync(request);

final ReceiveMessageRequest receiveRq = new ReceiveMessageRequest()
    .withMaxNumberOfMessages(1)
    .withQueueUrl(queueUrl);
final ReceiveMessageResult rx = bufferedSqs.receiveMessage(receiveRq);
```

Configurar o `AmazonSQSBufferedAsyncClient`

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);
```

Parâmetros de configuração `QueueBufferConfig`

Parâmetro	Valor padrão	Descrição
<code>longPoll</code>	<code>true</code>	Quando <code>longPoll</code> está definido como <code>true</code> , <code>AmazonSQSBufferedAsyncClient</code> tenta usar a sondagem longa ao consumir mensagens.
<code>longPollWaitTimeoutSeconds</code>	20 s	A quantidade máxima de tempo, em segundos, em que uma chamada <code>ReceiveMessage</code> é bloqueada no servidor

Parâmetro	Valor padrão	Descrição
		<p>aguardando as mensagens aparecerem na fila antes de retornar com um resultado de recebimento vazio.</p> <p>Note</p> <p>Quando a sondagem longa está desativada, essa configuração não tem efeito.</p>
<code>maxBatchOpenMs</code>	200ms	<p>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.</p> <p>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).</p> <p>Quando esse parâmetro é definido como 0, as solicitações enviadas não aguardam outras solicitações, desativando efetivamente o processamento em lotes.</p>
<code>maxBatchSize</code>	10 solicitações por lote	<p>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.</p> <p>Note</p> <p>10 solicitações por lote é o valor máximo permitido para o Amazon SQS.</p>

Parâmetro	Valor padrão	Descrição
<code>maxBatchSizeBytes</code>	256 KB	<p>O tamanho máximo de um lote de mensagens, em bytes, que o cliente tenta enviar ao Amazon SQS.</p> <p>Note</p> <p>256 KB é o valor máximo permitido para o Amazon SQS.</p>
<code>maxDoneReceiveBatches</code>	10 lotes	<p>O número máximo de lotes de recebimento que <code>AmazonSQSBufferedAsyncClient</code> pré-busca e armazena no lado do cliente.</p> <p>Quanto maior for a configuração, mais solicitações de recebimento poderão ser atendidas sem a necessidade de fazer uma chamada para o 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).</p> <p>Note</p> <p>o indica que toda a pré-busca de mensagens está desativada e as mensagens são consumidas apenas sob demanda.</p>
<code>maxInflightOutboundBatches</code>	5 lotes	<p>O número máximo de lotes de saída ativos que podem ser processados ao mesmo tempo.</p> <p>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 <code>AmazonSQSBufferedAsyncClient</code>.</p>

Parâmetro	Valor padrão	Descrição
<code>maxInflightReceiveBatches</code>	10 lotes	<p>O número máximo de lotes de recebimento ativos que podem ser processados ao mesmo tempo.</p> <p>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 <code>AmazonSQSBufferedAsyncClient</code>.</p> <p>Note</p> <p>o indica que toda a pré-busca de mensagens está desativada e as mensagens são consumidas apenas sob demanda.</p>
<code>visibilityTimeoutSeconds</code>	-1	<p>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.</p> <p>Note</p> <p>-1 indica que a configuração padrão foi selecionada para a fila. Não é possível configurar o tempo limite de visibilidade para 0.</p>

Aumento da taxa de transferência usando escalabilidade horizontal e processamento de ações em lotes

As filas do Amazon SQS podem fornecer taxa de transferência muito altas. As filas padrão oferecem suporte a um número quase ilimitado de chamadas de API por segundo, por ação de API (`SendMessage`, `ReceiveMessage` ou `DeleteMessage`). Se você usa o [agrupamento em lote](#) (p. 166), as filas FIFO oferecem suporte a até 3.000 transações por segundo, por método de API (`SendMessageBatch`, `ReceiveMessage` ou `DeleteMessageBatch`). As 3000 transações 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`).

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](#) (p. 172)
- [Processamento de ações em lotes](#) (p. 172)
- [Exemplo de Java funcional para operações únicas e solicitações em lote](#) (p. 173)

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 atingir a partir de um único thread através de uma única conexão. Por exemplo, se a média de latência 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 um único thread através de uma única conexão será em média 50 operações por segundo (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, certifique-se de que o cliente do Amazon SQS tenha conexões ou threads suficientes para oferecer suporte ao número de produtores e consumidores de mensagens simultâneos que enviarão solicitações e receberão respostas. Por exemplo, por padrão, instâncias do AWS SDK para Java [AmazonSQSClient](#) mantém, no máximo, 50 conexões para 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.

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 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 no buffer do Amazon SQS](#) (p. 167).

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

Prerequisites

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>
```

```
</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 2010-2021 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 as métricas de volume para mensagens enviadas, recebidas e excluídas. Você pode acessar essas métricas e outras por meio do [Monitoramento](#) para sua fila ou no [Console do CloudWatch](#).

Note

As métricas podem levar até 15 minutos após a fila começar para ficar disponíveis.

Recursos do Amazon SQS

A tabela a seguir lista os recursos relacionados que serão úteis à medida que você utilizar este serviço.

Recurso	Descrição
Referência da API do 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 noReferência de comandos da CLI da AWS	Descrições dos comandos da CLI da AWS que você pode usar para trabalhar com filas.
Regiões e endpoints	Informações sobre regiões do Amazon SQS 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.
Informações sobre o AWS Premium Support	A principal página da web para obter informações sobre o AWS Premium Support, um canal de suporte de resposta rápida e com atendimento individual, para ajudá-lo a desenvolver e executar aplicativos nos AWS Infrastructure Services.

Histórico de documentação

A tabela a seguir descreve as alterações importantes no Amazon Simple Queue Service Developer desde janeiro de 2019. Para receber notificações sobre atualizações dessa documentação, inscreva-se no [Feed RSS](#).

update-history-change	update-history-description	update-history-date
Alta taxa de transferência para mensagens em filas FIFO está disponível na versão de pré-visualização	A alta taxa de transferência para filas FIFO do Amazon SQS está na versão de teste e sujeito a alterações. Esse recurso fornece um número maior de transações 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 de console do Amazon SQS	Para simplificar fluxos de trabalho de desenvolvimento e produção, o console do Amazon SQS tem um Nova experiência do usuário .	8 de julho de 2020
O Amazon SQS oferece suporte à paginação para listQueues e listDeadLetterSourceQueues	É possível especificar o número máximo de resultados a serem retornados de uma solicitação listQueues ou listDeadLetterSourceQueues .	22 de junho de 2020
O Amazon SQS oferece suporte a métricas do Amazon CloudWatch de 1 minuto em todas as regiões da AWS, exceto nas regiões do AWS GovCloud (EUA)	A métrica de um minuto do CloudWatch do Amazon SQS está disponível em todas as regiões, exceto as regiões da AWS GovCloud (EUA).	9 de janeiro de 2020
O Amazon SQS oferece suporte a métricas CloudWatch de 1 minuto	No momento, a métrica de um minuto do CloudWatch para Amazon SQS está disponível apenas nas seguintes regiões: Leste dos EUA (Ohio), Europa (Irlanda), Europa (Estocolmo) e Ásia-Pacífico (Cingapura)	25 de novembro de 2019
Os gatilhos do AWS Lambda para filas FIFO do Amazon SQS estão disponíveis	Você pode configurar as mensagens que chegam em uma fila FIFO como um gatilho de função do Lambda.	25 de novembro de 2019
A criptografia no lado do servidor (SSE) para o Amazon SQS está disponível nas regiões da China	O SSE para o Amazon SQS está disponível nas regiões da China.	13 de novembro de 2019

As filas FIFO estão disponíveis na região do Oriente Médio (Bahrein)	As filas FIFO estão disponíveis na região do Oriente Médio (Bahrein).	10 de outubro de 2019
Endpoints da Amazon Virtual Private Cloud (Amazon VPC) para Amazon SQS estão disponíveis nas regiões AWS GovCloud (EUA-Leste) e AWS GovCloud (EUA-Oeste)	Você pode enviar mensagens para as filas do Amazon SQS da Amazon VPC nas regiões AWS GovCloud (EUA-Leste) e AWS GovCloud (EUA-Oeste).	5 de setembro de 2019
O Amazon SQS permite a solução de problemas de filas usando o 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 <code>MessageSystemAttribute</code> parâmetro de solicitação (que permite enviar cabeçalhos de rastreamento de X-Ray por meio do Amazon SQS) para o <code>SendMessage</code> e <code>SendMessageBatch</code> Operações de API, o <code>AWSTraceHeaderAttribute</code> para o <code>ReceiveMessage</code> a operação da <code>APIMessageSystemAttributeValue</code> Tipo de dados.	28 de agosto de 2019
É possível marcar 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 as tags dele. Além disso, o Amazon SQS oferece suporte ao <code>aws:TagKey</code> e <code>aws:RequestTag</code> Chaves AWS Identity and Access Management (IAM).	22 de agosto de 2019
O Temporary Queue Client para Amazon SQS agora 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 Cliente de Filas tempor para criar filas temporárias gerenciadas por aplicativos de alto débito, econômicas e econômicas.	25 de julho de 2019
O SSE for Amazon SQS está disponível na região AWS GovCloud (Leste dos EUA)	A criptografia no lado do servidor (SSE) para o Amazon SQS está disponível na região AWS GovCloud (Leste dos EUA)	20 de junho de 2019

As filas FIFO estão disponíveis nas regiões Ásia-Pacífico (Hong Kong), China (Pequim), AWS GovCloud (EUA-Leste) e AWS GovCloud (EUA-Oeste)	As filas FIFO estão disponíveis nas regiões Ásia-Pacífico (Hong Kong), China (Pequim), AWS GovCloud (EUA-Leste) e AWS GovCloud (EUA-Oeste).	15 de maio de 2019
As políticas de VPC endpoint Amazon 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 da 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 no Leste dos EUA (Ohio), Leste dos EUA (Virgínia), Oeste dos EUA (Oregon), Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Tóquio), Europa (Frankfurt), Europa (Irlanda), Europa (Londres), Europa (Paris) e regiões América do Sul (São Paulo).	7 de fevereiro de 2019

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o [Glossário da AWS](#) na Referência geral da AWS.

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.