



Hands-on tutorials

Send Messages Between Distributed Applications



Send Messages Between Distributed Applications: Hands-on tutorials

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Send Messages Between Distributed Applications **i**

Overview 1

Implementation 1

Conclusion 14

Send Messages Between Distributed Applications

Cost to Complete	Free Tier
Services Used	Amazon SQS
Sending Messages on AWS Requires an Account	AWS Free Tier includes 1,000,000 requests of Amazon Simple Queue Service. View AWS Free Tier Details » Create a Free Account in Minutes

Overview

In this tutorial, you will learn how to set up asynchronous messaging with [Amazon Simple Queue Service \(Amazon SQS\)](#). Amazon SQS is the AWS service that allows application components to communicate in the cloud. You will use the Amazon SQS console to create and configure a message queue, send a message, receive and delete that message, and then delete the queue.

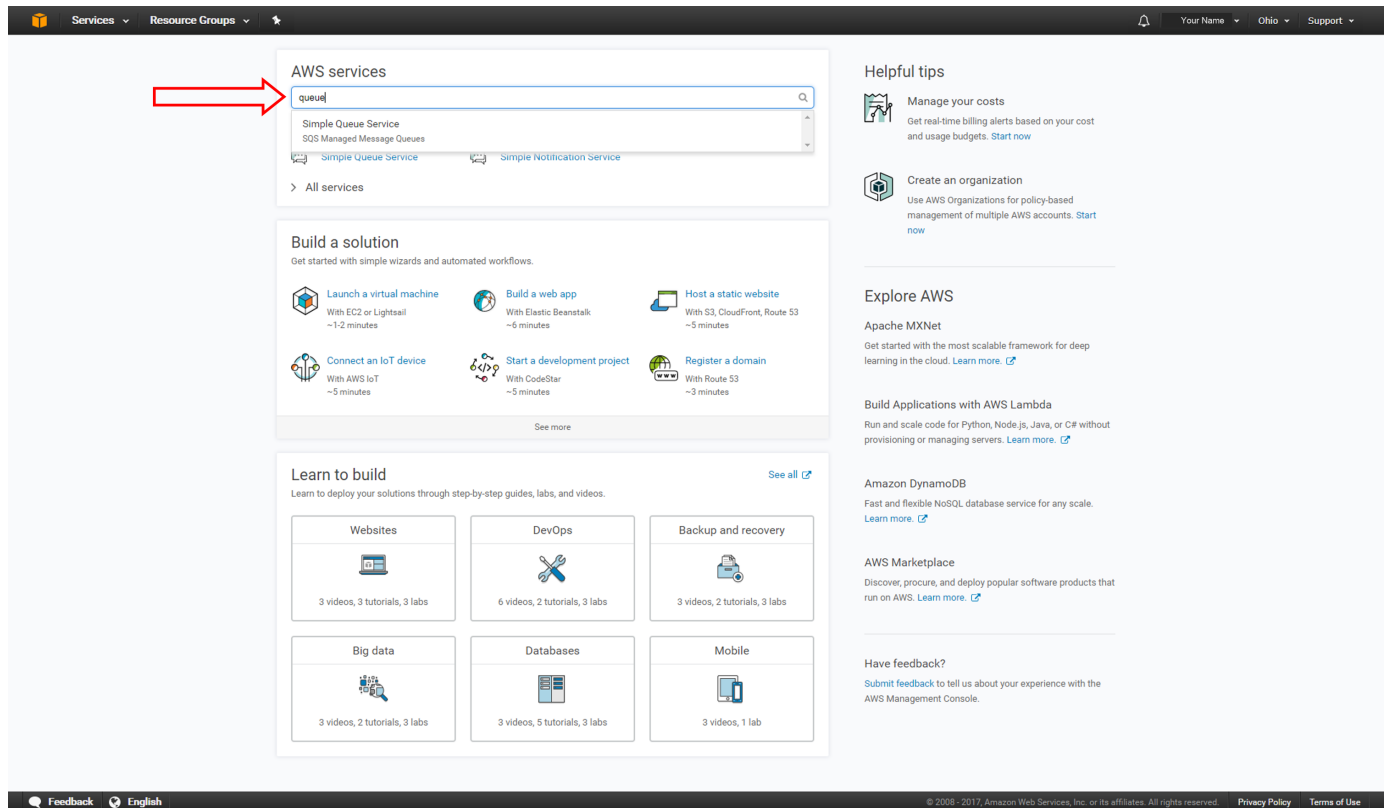
The AWS services you use in this tutorial are within the [AWS Free Tier](#).

Implementation

Step 1: Open the Amazon SQS console

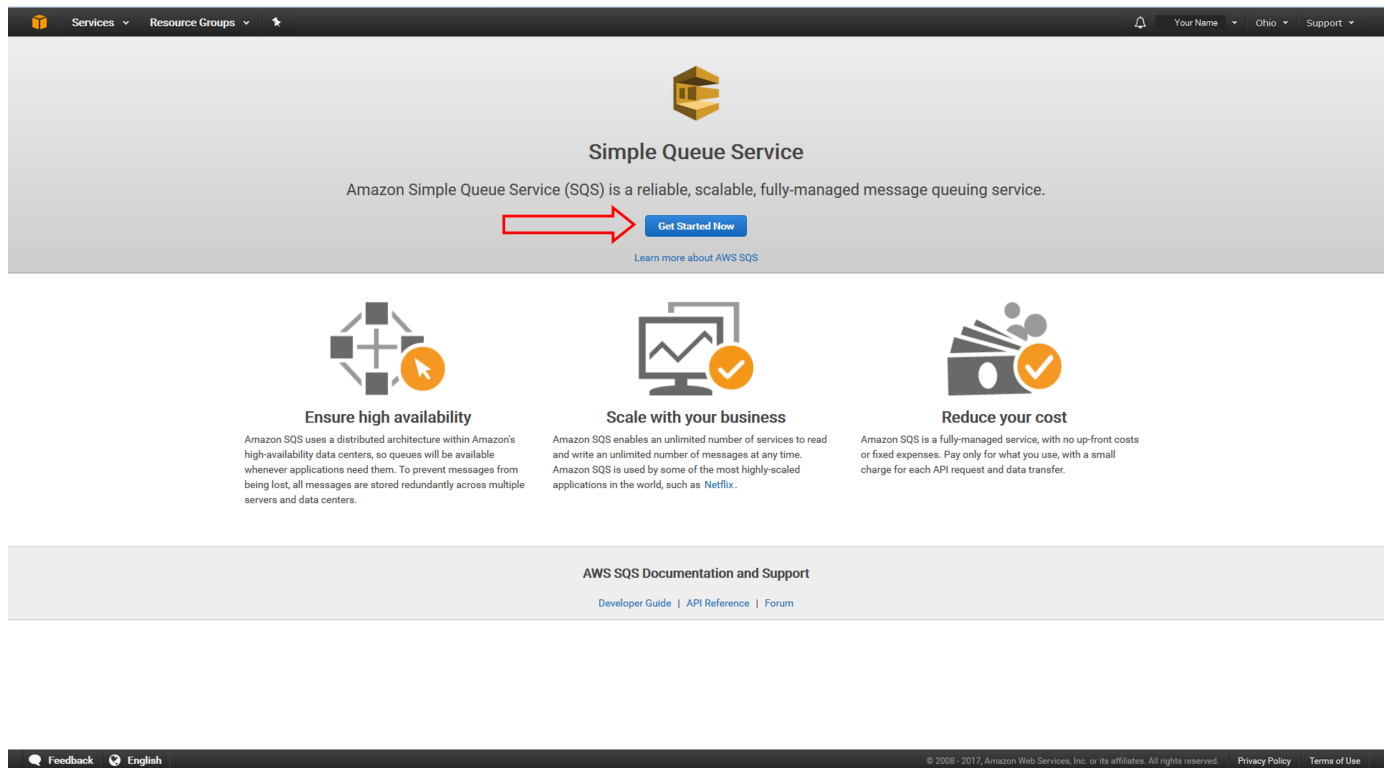
1. Launch the AWS Management Console

When you [click here](#), the AWS Management Console will open in a new browser window, so you can keep this step-by-step guide open. When the screen loads, enter your user name and password to get started. Then type **queue** in the search bar and select **Simple Queue Service** to open the console.



2. Start the Amazon SQS Console

If the SQS console landing page appears, as shown on by the screenshot, click **Get Started Now**. If you don't see this page, skip to the next step.



Services Resource Groups

Simple Queue Service

Amazon Simple Queue Service (SQS) is a reliable, scalable, fully-managed message queuing service.

[Get Started Now](#)

[Learn more about AWS SQS](#)

Ensure high availability
Amazon SQS uses a distributed architecture within Amazon's high-availability data centers, so queues will be available whenever applications need them. To prevent messages from being lost, all messages are stored redundantly across multiple servers and data centers.

Scale with your business
Amazon SQS enables an unlimited number of services to read and write an unlimited number of messages at any time. Amazon SQS is used by some of the most highly-scaled applications in the world, such as [Netflix](#).

Reduce your cost
Amazon SQS is a fully-managed service, with no up-front costs or fixed expenses. Pay only for what you use, with a small charge for each API request and data transfer.

[AWS SQS Documentation and Support](#)

[Developer Guide](#) | [API Reference](#) | [Forum](#)

Feedback English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Step 2: Create an Amazon SQS Queue

In this step, you will create and configure an Amazon SQS queue. A queue is a reliable, highly-scalable buffer that stores messages as they travel between distributed applications or microservices. Queues help to decouple applications, connect microservices, batch tasks, or store notifications.

Our use case for this tutorial will simulate the storage of incoming orders from an e-commerce application.

1. Enter a queue name

First, we will create a simple queue that stores orders that are placed on the store. Enter **Orders** in the **Queue Name** field.

Services

Resource Groups

Your NameOhioSupport

Create New Queue

What do you want to name your queue?

Queue Name [?]

Orders

Region [?] US East (Ohio)

What type of queue do you need?

Standard Queue

FIFO Queue

High Throughput: Standard queues have nearly-unlimited transactions per second (TPS).

At-Least-Once Delivery: A message is delivered at least once, but occasionally more than one copy or a message is delivered.

Best-Effort Ordering: Occasionally, messages are delivered in an order different from which they were sent.

Send data between applications when the throughput is important, for example:

- Decouple live user requests from intensive background work: let users upload media while resizing or encoding it.
- Allocate tasks to multiple worker nodes: process a high number of credit card validation requests.
- Batch messages for future processing: schedule multiple entries to be added to a database.

First-In-First-Out Delivery: The order in which messages are sent and received is strictly preserved.

Exactly-Once Processing: A message is guaranteed to be delivered at least once, but all duplicates of the message are removed.

Limited Throughput: 300 transactions per second (TPS).

Send data between applications when the order of events is important, for example:

- Ensure that user-entered commands are executed in the right order.
- Display the correct product price by sending price modifications in the right order.
- Prevent a student from enrolling in a course before registering for an account.

FeedbackEnglish

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy PolicyTerms of Use

2. Choose a queue type

For this tutorial, we do not require strict ordering, so we won't make any changes to the queue type. Leave **Standard Queue** selected.

Services

Resource Groups

Your NameOhioSupport

Create New Queue

What do you want to name your queue?

Queue Name [?]

Orders

Region [?] US East (Ohio)

What type of queue do you need?

Standard Queue

FIFO Queue

High Throughput: Standard queues have nearly-unlimited transactions per second (TPS).

At-Least-Once Delivery: A message is delivered at least once, but occasionally more than one copy or a message is delivered.

Best-Effort Ordering: Occasionally, messages are delivered in an order different from which they were sent.

Send data between applications when the throughput is important, for example:

- Decouple live user requests from intensive background work: let users upload media while resizing or encoding it.
- Allocate tasks to multiple worker nodes: process a high number of credit card validation requests.
- Batch messages for future processing: schedule multiple entries to be added to a database.

First-In-First-Out Delivery: The order in which messages are sent and received is strictly preserved.

Exactly-Once Processing: A message is guaranteed to be delivered at least once, but all duplicates of the message are removed.

Limited Throughput: 300 transactions per second (TPS).

Send data between applications when the order of events is important, for example:

- Ensure that user-entered commands are executed in the right order.
- Display the correct product price by sending price modifications in the right order.
- Prevent a student from enrolling in a course before registering for an account.

FeedbackEnglish

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy PolicyTerms of Use

3. Create the queue

You can configure your queue to modify settings such as retention period, maximum message size and delivery delays. For this tutorial, we will keep the default parameters. Choose **Quick-Create Queue**.

The screenshot shows the AWS Management Console interface for creating a new queue. At the top, there's a navigation bar with 'Services' and 'Resource Groups'. Below that, a search bar contains 'Orders' and the region is set to 'US East (Ohio)'. The main heading is 'What type of queue do you need?'. There are two tabs: 'Standard Queue' (selected) and 'FIFO Queue'. The 'Standard Queue' section describes its characteristics: High Throughput (nearly-unlimited TPS), At-Least-Once Delivery (messages may be delivered more than once), and Best-Effort Ordering (messages may be delivered in a different order). It includes a diagram showing a non-sequential flow of messages (1, 5, 4, 2, 3) and a list of use cases like decoupling live user requests from intensive background work. The 'FIFO Queue' section describes its characteristics: First-In-First-Out Delivery (strictly preserved order), Exactly-Once Processing (guaranteed at least once delivery with duplicate removal), and Limited Throughput (300 TPS). It includes a diagram showing a sequential flow of messages (1, 2, 3, 4, 5) and a list of use cases like ensuring user-entered commands are executed in the right order. At the bottom, there are three buttons: 'Cancel', 'Configure Queue', and 'Quick-Create Queue'. A red arrow points to the 'Quick-Create Queue' button. Below the buttons, a footer contains 'Feedback', 'English', and copyright information.

4. Verify queue creation

Your new queue is created and selected in the queue list.

The screenshot shows the AWS Management Console interface for Amazon SQS. At the top, there are navigation tabs for 'Services' and 'Resource Groups'. Below this, a 'Create New Queue' button and a 'Queue Actions' dropdown are visible. A search bar labeled 'Filter by Prefix: Enter Text...' is present. A table lists the queues, with the 'Orders' queue highlighted in blue and a red border. The table columns are: Name, Queue Type, Content-Based Deduplication, Messages Available, Messages in Flight, and Created. The 'Orders' queue has a 'Standard' type, 'N/A' for deduplication, 0 messages available, 0 in flight, and was created on 2017-08-10 11:03:28 GMT-07:00.

Below the table, the 'Details' tab for the 'Orders' queue is selected. It shows the following information:

- Name: Orders
- URL: <https://sqs.us-east-2.amazonaws.com/585534329928/Orders>
- ARN: [arn:aws:sqs:us-east-2:585534329928:Orders](#)
- Created: 2017-08-10 11:03:28 GMT-07:00
- Last Updated: 2017-08-10 11:03:28 GMT-07:00
- Delivery Delay: 0 seconds
- Queue Type: Standard
- Content-Based Deduplication: N/A
- Default Visibility Timeout: 30 seconds
- Message Retention Period: 4 days
- Maximum Message Size: 256 KB
- Receive Message Wait Time: 0 seconds
- Messages Available (Visible): 0
- Messages in Flight (Not Visible): 0
- Messages Delayed: 0

At the bottom of the console, there are links for 'Feedback', 'English', and a footer with copyright information: '© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.

Step 3: Send messages to the Queue

Once you have created your queue, it is ready to receive messages from the online store that capture the details of each new order.

1. Send a message

Your queue is already selected in the list. From **Queue Actions**, select **Send a Message**. The **Send a Message to Orders** dialog box is displayed.

The screenshot shows the AWS Management Console interface for an Amazon SQS queue named 'Orders'. The 'Queue Actions' dropdown menu is open, and the 'Send a Message' option is highlighted with a red arrow. The queue details are displayed below, showing various configuration parameters.

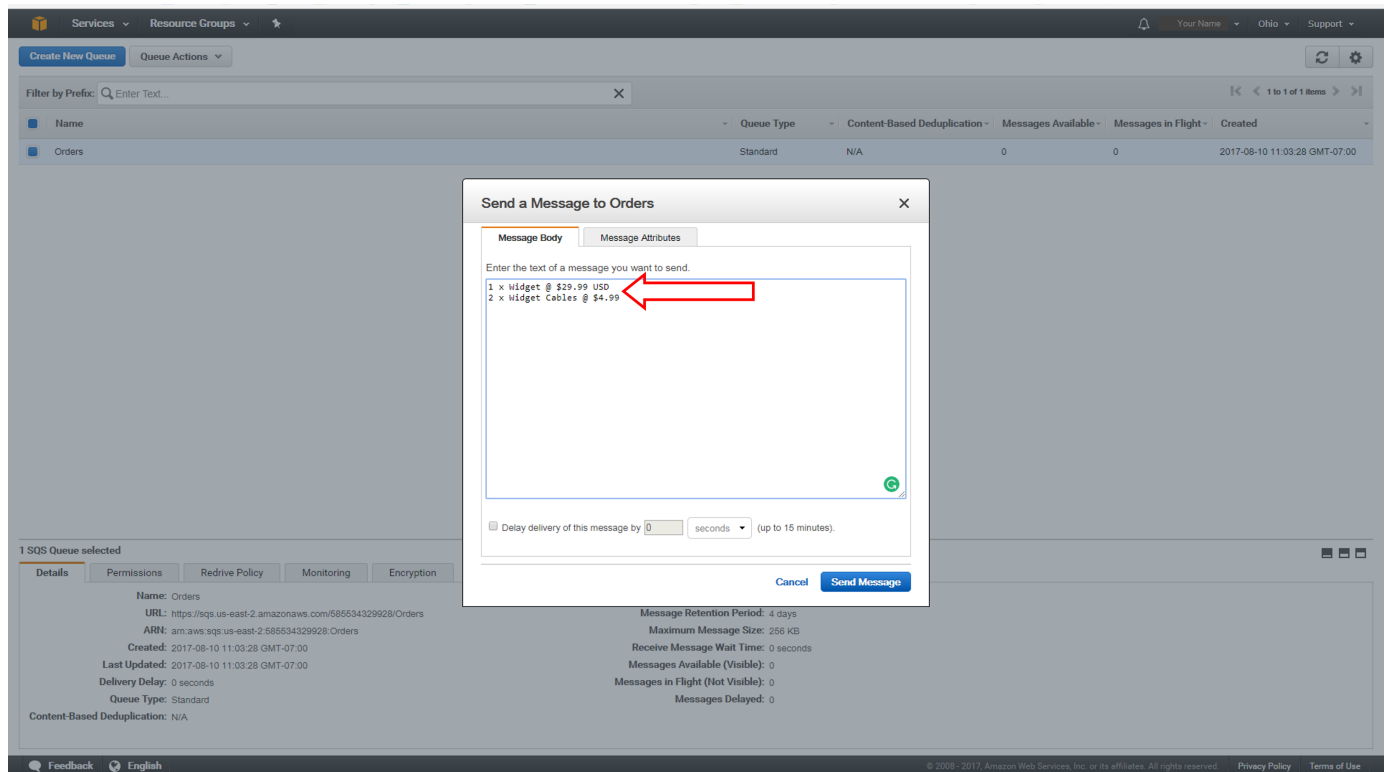
Queue Name	Queue Type	Content-Based Deduplication	Messages Available	Messages in Flight	Created
Orders	Standard	N/A	0	0	2017-08-10 11:03:28 GMT-07:00

1 SQS Queue selected				
Details	Permissions	Redrive Policy	Monitoring	Encryption
Name: Orders URL: https://sqs.us-east-2.amazonaws.com/585534329928/Orders ARN: arn:aws:sqs:us-east-2:585534329928:Orders Created: 2017-08-10 11:03:28 GMT-07:00 Last Updated: 2017-08-10 11:03:28 GMT-07:00 Delivery Delay: 0 seconds Queue Type: Standard Content-Based Deduplication: N/A Default Visibility Timeout: 30 seconds Message Retention Period: 4 days Maximum Message Size: 256 KB Receive Message Wait Time: 0 seconds Messages Available (Visible): 0 Messages in Flight (Not Visible): 0 Messages Delayed: 0				

2. Enter message content

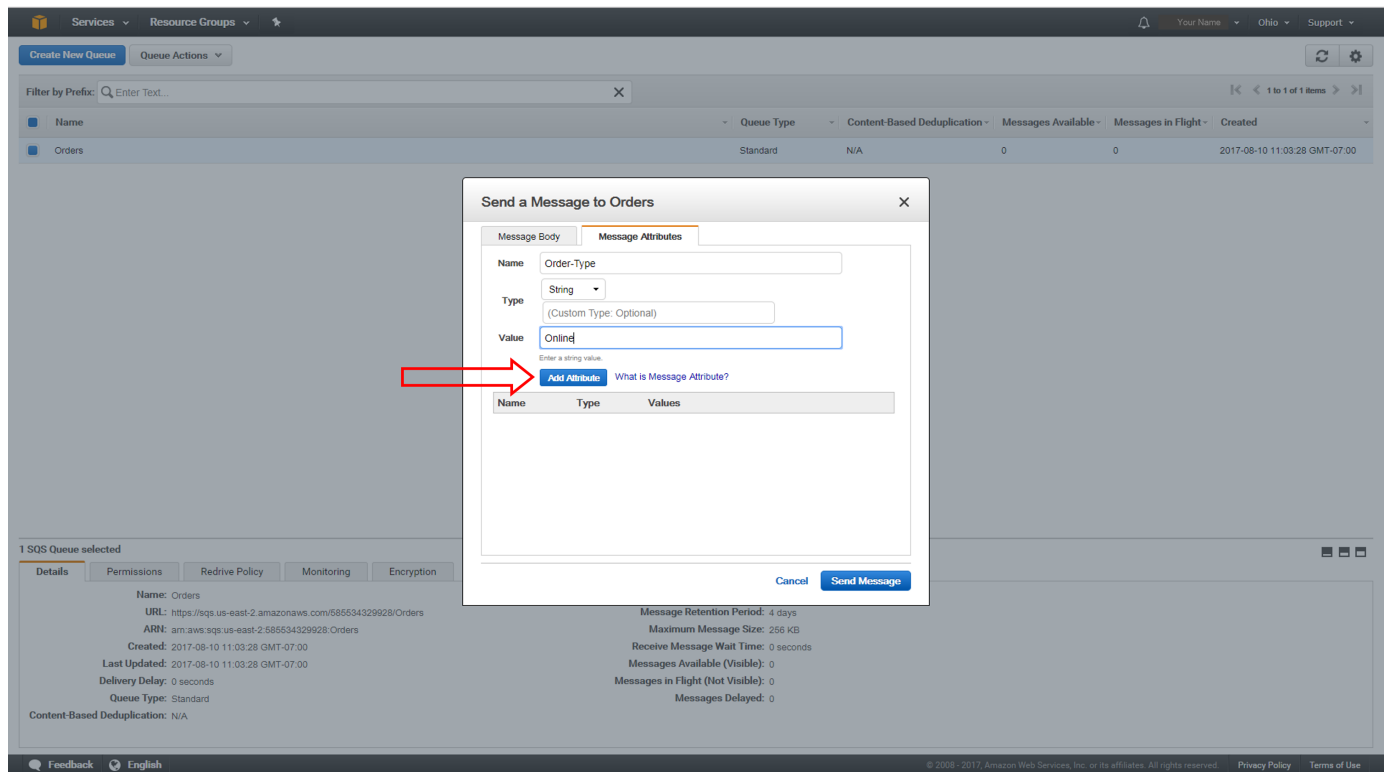
The **Send a Message to Orders** dialog box is displayed. On the **Message Body** tab, enter the following text to represent a sample order:

1 x Widget @ \$29.99 USD 2 x Widget Cables @ \$4.99



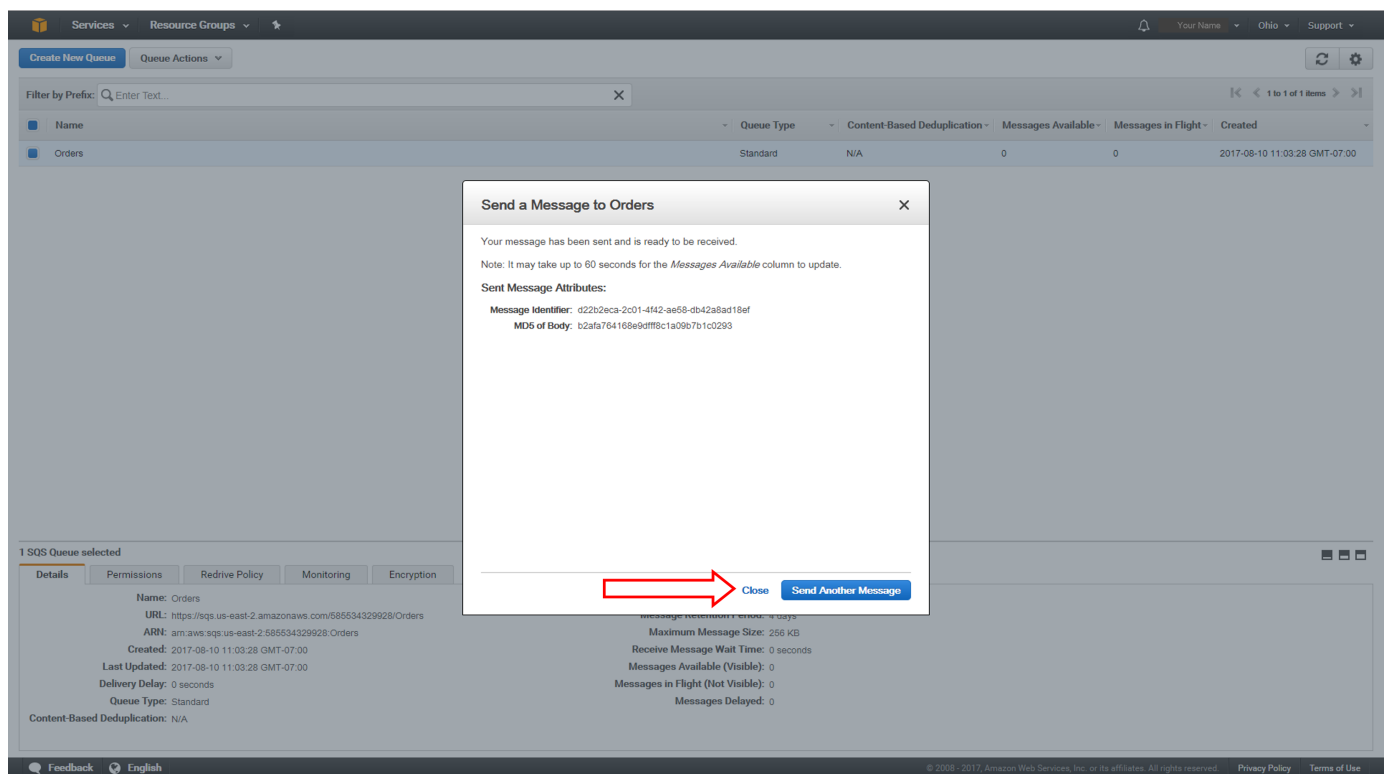
3. Enter message attributes

Select the **Message Attributes** tab to add some optional metadata about this message for easy processing. Let's add an order type to the order. Enter **Order-Type** in the **Name** field, **String** in the **Type** field, and **Online** in the **Value** field. Click **Add Attribute**.



4. Send message

To send the message immediately, click **Send Message**. Confirmation that your message was sent is displayed in the **Send a Message to Orders** dialog box. Click **Close**.

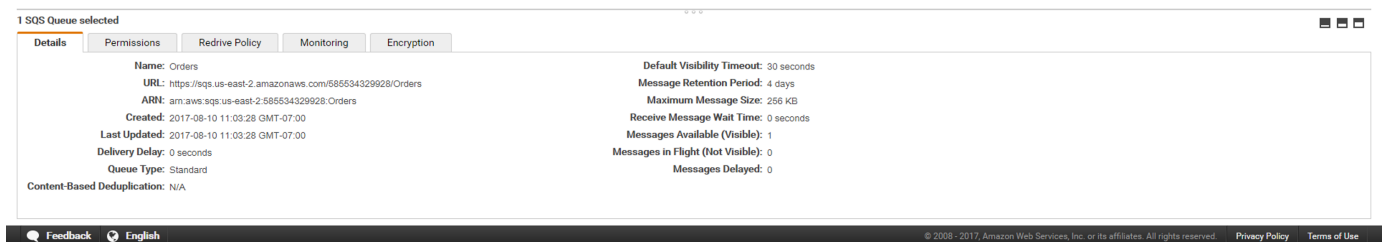
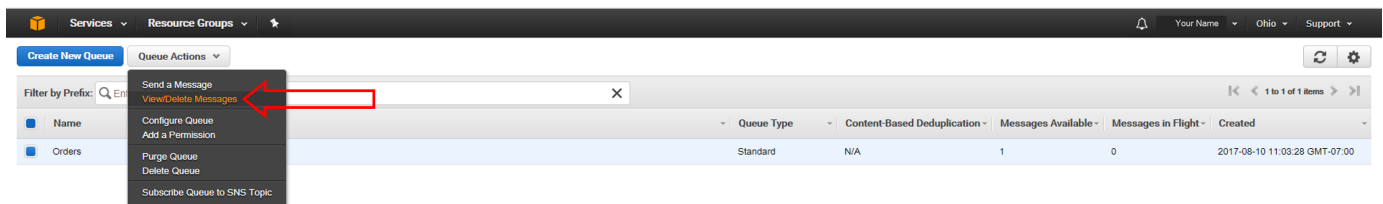


Step 4: Retrieve and Delete a Message

After you send a message to a queue, another application can consume it from the queue and do something with it. In this example, you will simply retrieve the message to view the order, and then delete it.

1. View or delete messages

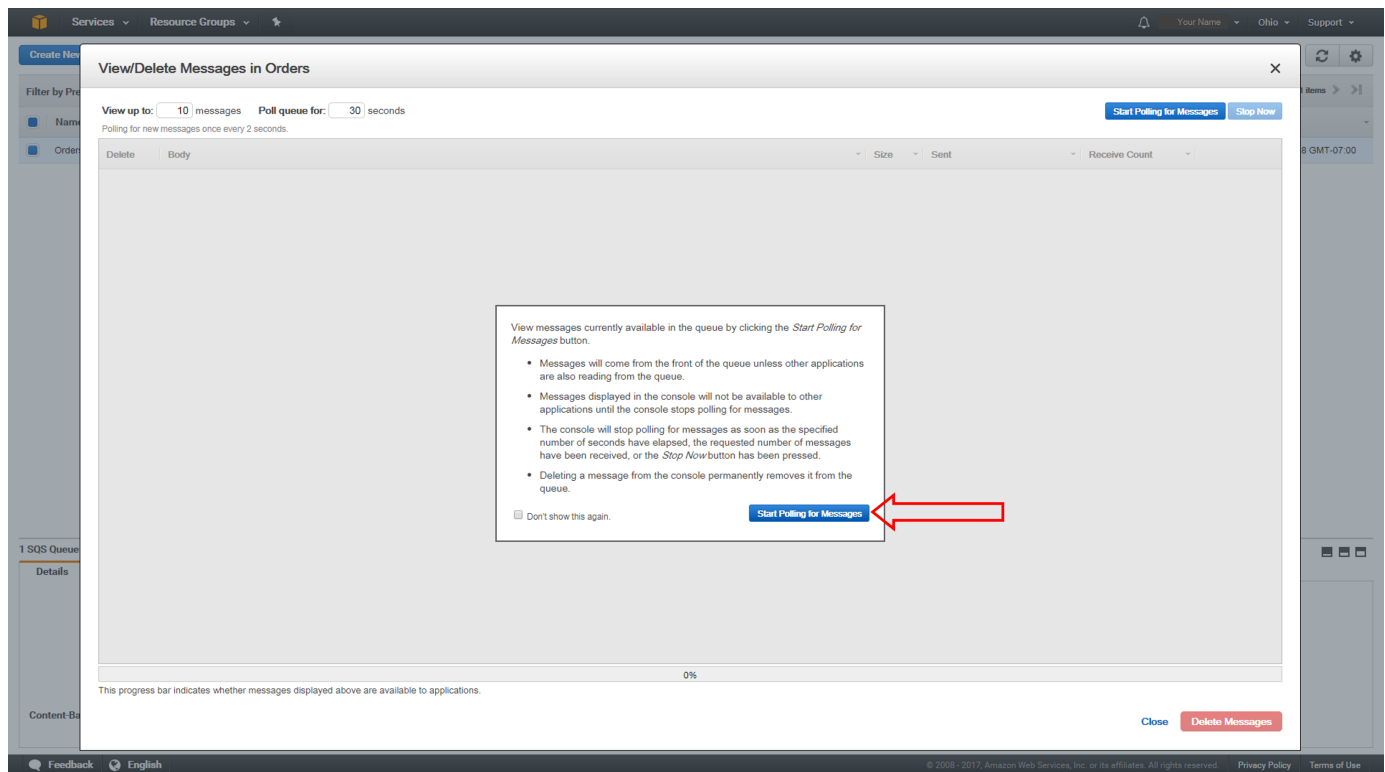
Ensure that your **Orders** queue is selected in the queue list. Next, from **Queue Actions**, select **View/Delete Messages**.



2. Poll for messages

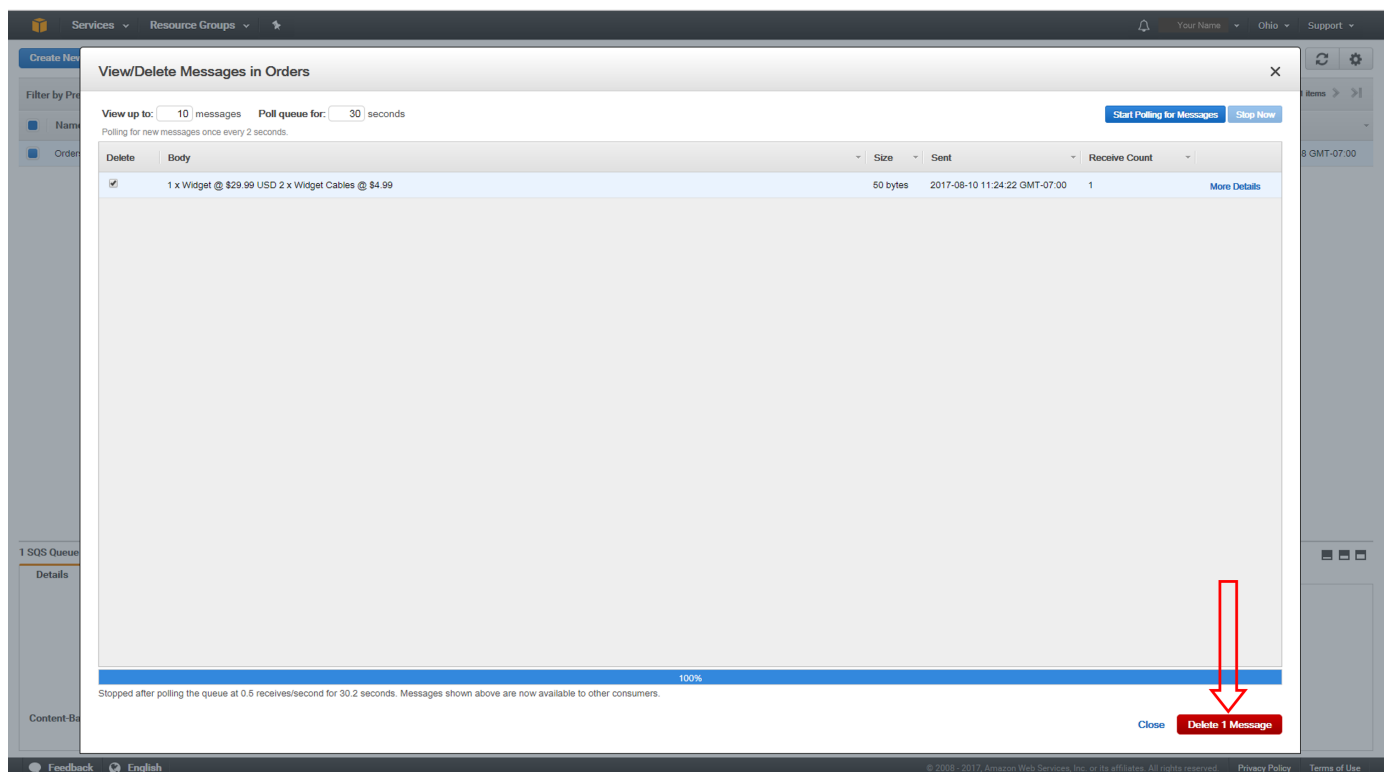
The **View/Delete Messages in Orders** dialog box is displayed. When you request a message from a queue, you don't specify request a specific message. Instead, you specify the maximum number of messages (up to 10) that you want to retrieve.

Click **Start Polling for messages** to retrieve messages from the queue.



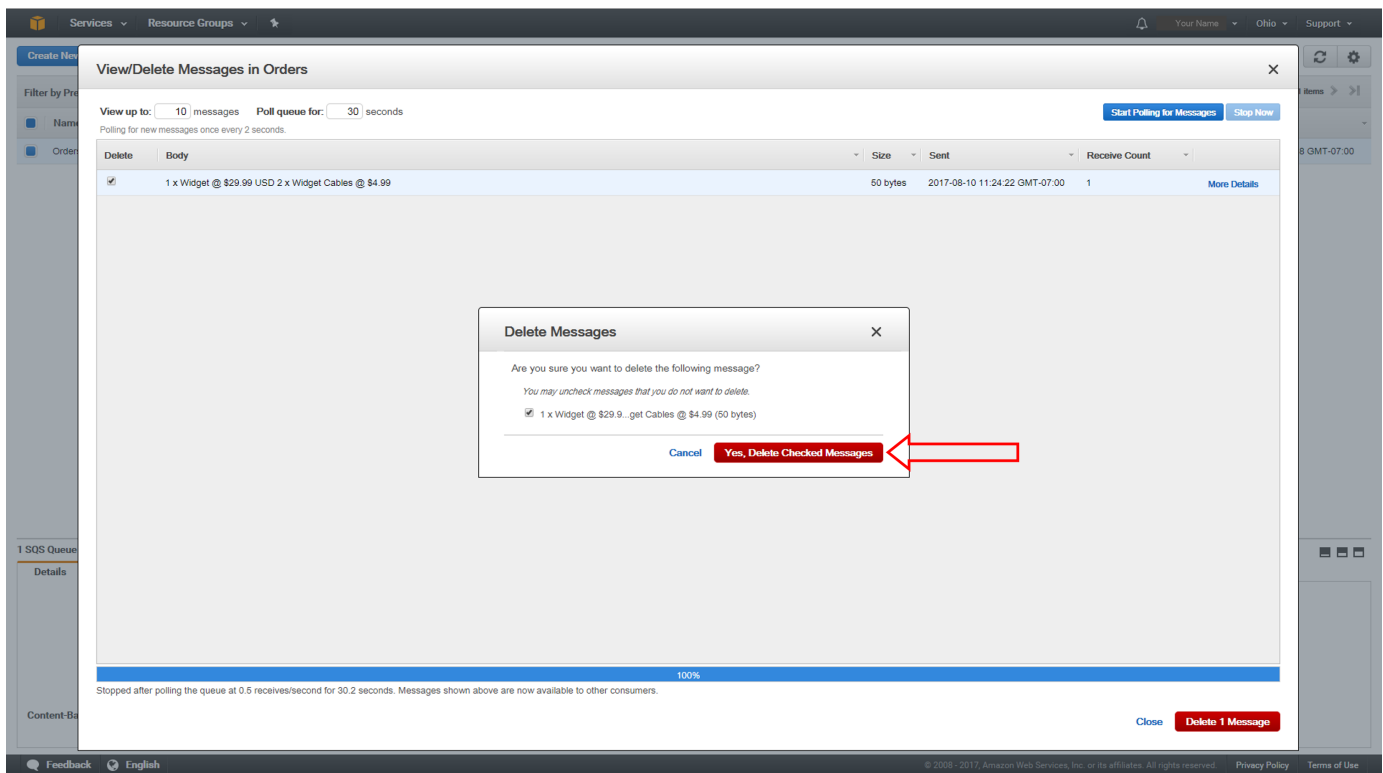
3. Select and delete the message

Once a consumer has received and processed a message, it can be deleted from the queue. Select the message that you want to delete and then choose **Delete 1 Message**.



4. Confirm deletion

The **Delete Messages** dialog box is displayed. Check the box next to the message and click **Yes, Delete Checked Messages**. The selected message is deleted. Choose **Close**.



Step 5: Clean Up Resources

When you no longer need to use an Amazon SQS queue, we recommend that you delete the queue as a best practice.

1. Select and delete queue

In the queue list, select the **Orders** queue. Then, from **Queue Actions**, select **Delete Queue**.

The screenshot shows the AWS IAM console interface. At the top, there's a navigation bar with 'Services', 'Resource Groups', and a search icon. Below this, a 'Queue Actions' dropdown menu is open, showing options like 'Send a Message', 'View/Delete Messages', 'Configure Queue', 'Add a Permission', 'Purge Queue', 'Delete Queue' (highlighted with a red arrow), and 'Subscribe Queue to SNS Topic'. The main table lists one queue named 'Orders' with a 'Standard' type, 'N/A' for content-based deduplication, and 0 messages available and in flight. Below the table, the 'Details' tab for the 'Orders' queue is expanded, showing metadata such as Name, URL, ARN, Created, Last Updated, Delivery Delay, Queue Type, Content-Based Deduplication, Default Visibility Timeout, Message Retention Period, Maximum Message Size, Receive Message Wait Time, Messages Available (Visible), Messages in Flight (Not Visible), and Messages Delayed.

2. Confirm deletion

The **Delete Queues** dialog box is displayed. You can still delete your queue, even though you still have messages in it. Choose **Yes, Delete Queue**. The queue is deleted.

This screenshot shows the same AWS IAM console interface as the previous one, but with a 'Delete Queues' dialog box open in the center. The dialog box asks, 'Are you sure you want to delete the following queue, and any messages left in it?' and lists 'Orders - contains no messages.' Below the list are two buttons: 'Cancel' and 'Yes, Delete Queue' (highlighted with a red arrow). The background shows the 'Orders' queue details, which are dimmed.

Conclusion

You have created your first Amazon Simple Queue Service (Amazon SQS) message queue, sent messages to your queue, retrieved and deleted messages, and then deleted the queue. You are now ready to use Amazon SQS queues to store and move data between distributed application components and microservices.